SQL Database Basics

المرحلة الثانية/ الفصل الدراسي الاول

قسم تقنيات أنظمة الحاسوب

المعهد التقني في البصرة

اعداد

المدرس المساعد زينب محمد جوار

السنة الدراسية: ٢٠٢٤–٢٠٢٥

What is a Database?

ما هي قاعدة البيانات؟

A database is like a digital warehouse where data is stored, organized, and managed efficiently. It acts as a central hub for information, making it easier to access and analyze data.

تعتبر قاعدة البيانات بمثابة مستودع رقمي حيث يتم تخزين البيانات وتنظيمها وإدارتها بكفاءة. تعمل كمجمع مركزي للمعلومات، مما يجعل الوصول إلى البيانات وتحليلها أسهل.

Key Components of a Database:

- **Data:** Data is the facts or raw material (raw information) stored in a database, such as customer details, product information, or financial records. It can be in different formats like text, numbers, dates, or images. To understand these data, they must be translated or interpreted to become information. Information is the meaning that is given to the data through interpretation appropriately.
- **البيانات:** البيانات هي الحقائق أو المواد الخام (المعلومات الخام) المخزنة في قاعدة البيانات، مثل تفاصيل العملاء أو معلومات المنتج أو السجلات المالية. ويمكن أن تكون في تنسيقات مختلفة مثل النص أو الأرقام أو التواريخ أو الصور. لفهم هذه البيانات، يجب ترجمتها أو تفسيرها لتصبح معلومات. المعلومات هي المعنى الذي يُعطى للبيانات من خلال التفسير المناسب.

- **Tables:** Tables are like virtual spreadsheets within a database. They have rows and columns, where each row represents a specific record or instance, and each column represents a particular piece of data. For example, a customer table may have columns like ID, Name, Address, and Contact.
- **الجداول:** الجداول تشبه جداول البيانات الافتراضية داخل قاعدة البيانات. تحتوي على صفوف وأعمدة، حيث يمثل كل صف سجلاً أو مثيلاً محددًا، ويمثل كل عمود جزءًا معينًا من البيانات. على سبيل المثال، قد يحتوي جدول العملاء على أعمدة مثل المعرف والاسم والعنوان وجهة الاتصال.

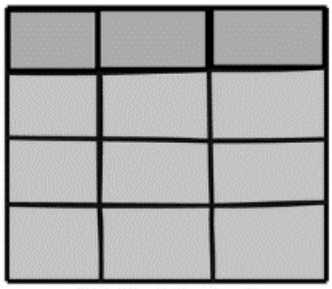
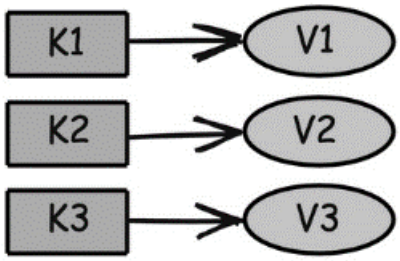
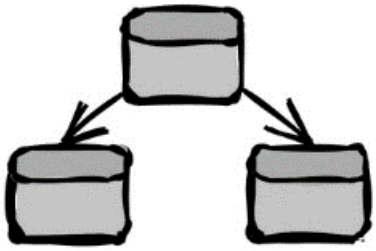
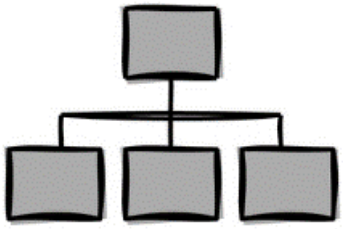
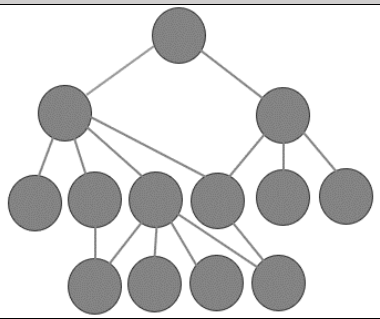
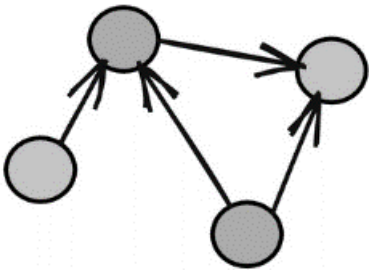
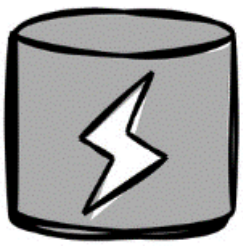
- **Relationships:** Relationships define how tables are connected within a database. They establish associations based on shared data elements. For instance, a customer's ID in one table can be linked to their orders in another. This helps maintain data consistency and enables efficient data retrieval.
- **العلاقات:** تحدد العلاقات كيفية ربط الجداول داخل قاعدة البيانات. فهي تنشئ ارتباطات بناءً على عناصر البيانات المشتركة. على سبيل المثال، يمكن ربط معرف العميل في جدول واحد بطلباته في جدول آخر. يساعد هذا في الحفاظ على اتساق البيانات وتمكين استرداد البيانات بكفاءة.

- **Queries:** Queries are like search commands that allow users to extract specific data from the database. Users can search, filter, and sort data based on the specified criteria. For example, a query can be used to find all customers who purchased in the last month.

- **الاستعلامات:** الاستعلامات تشبه أوامر البحث التي تسمح للمستخدمين باستخراج بيانات معينة من قاعدة البيانات. يمكن للمستخدمين البحث عن البيانات وتصفيتها وفرزها بناءً على المعايير المحددة. على سبيل المثال، يمكن استخدام الاستعلام للعثور على جميع العملاء الذين قاموا بالشراء في الشهر الماضي.

Types of Databases:

Here are some common types of databases:

Relational Databases (RDBMS)	NoSQL Databases
	
Object-Oriented Databases	**Hierarchical Databases**
	
Network Databases	**Graph Databases**
	
In-Memory Databases	**Time-Series Databases**
	

DBMS and RDBMS:

DBMS (Database Management System) and RDBMS (Relational Database Management System) are software systems used to manage databases; they have different qualities.

DBMS(نظام إدارة قواعد البيانات) و RDBMS(نظام إدارة قواعد البيانات العلائقية) عبارة عن أنظمة برمجية تستخدم لإدارة قواعد البيانات، ولها صفات مختلفة.

Why required DBMS?

A DBMS is required to efficiently manage an organization's data flow. It handles tasks such as inserting data into the database and retrieving data from it. The DBMS ensures the consistency and integrity of the data, as well as the speed at which data can be accessed.

Examples: XML Window Registry, Forxpro, dbaseIIIplus etc.

لماذا نحتاج إلىDBMS ؟

يعد نظام إدارة قواعد البيانات ضروريًا لإدارة تدفق البيانات في المؤسسة بكفاءة. فهو يتعامل مع مهام مثل إدخال البيانات في قاعدة البيانات واسترجاع البيانات منها. ويضمن نظام إدارة قواعد البيانات اتساق البيانات وسلامتها، بالإضافة إلى السرعة التي يمكن بها الوصول إلى البيانات.

أمثلة: XML Window Registry، Forxpro، dbaseIIIplus وما إلى ذلك.

Why required RDMS?

Similarly, an RDBMS is required when we want to manage data relationally, using tables and relationships. It helps in reducing data duplication and maintaining the integrity of the database. RDBMS ensures that data is stored in a structured manner, allowing for efficient querying and retrieval.

Examples: PostgreSQL, MySQL, Oracle, Microsoft Access, SQL Server etc.

لماذا نحتاج إلىRDMS ؟

وبالمثل، نحتاج إلى نظام إدارة قواعد البيانات العلائقية عندما نريد إدارة البيانات بشكل علائقي، باستخدام الجداول والعلاقات. فهو يساعد في تقليل تكرار البيانات والحفاظ على سلامة قاعدة البيانات. ويضمن نظام إدارة قواعد البيانات العلائقية تخزين البيانات بطريقة منظمة، مما يسمح بالاستعلام والاسترجاع بكفاءة.

أمثلة: PostgreSQL، MySQL، Oracle، Microsoft Access، SQL Server وما إلى ذلك.

What is the Key?

In RDBMS systems, keys are fields that take part in the following operations on tables:

- To establish connections between two tables.
- To keep a table’s individuality.
- To maintain accurate and consistent data in the database.
- Possibly speed up data retrieval by enabling indexes on column (s).

في أنظمة إدارة قواعد البيانات العلائقية، المفاتيح هي حقول تشارك في العمليات التالية على الجداول:

- لإنشاء اتصالات بين جدولين.
- للحفاظ على خصوصية الجدول.
- للحفاظ على دقة البيانات وتناسقها في قاعدة البيانات.
- ربما تسريع استرجاع البيانات من خلال تمكين الفهارس على العمود (الأعمدة).

The following list includes the many key types that SQL Server supports:

1. Candidate Key
2. Primary Key
3. Unique Key
4. Alternate Key
5. Composite Key
6. Super Key
7. Foreign Key

What is a Primary Key?

A primary key is a single column value used to identify a database record uniquely.

ما هو المفتاح الأساسي؟

المفتاح الأساسي هو قيمة عمود واحد يستخدم لتحديد سجل قاعدة البيانات بشكل فريد.

primary key has the following attributes:

- A primary key cannot be NULL
- A primary key value must be unique
- The primary key values should rarely be changed
- The primary key must be given a value when a new record is inserted.

لديها السمات التالية:

- لا يمكن أن يكون المفتاح الأساسي فارغًا
- يجب أن تكون قيمة المفتاح الأساسي فريدة
- نادراً ما يتم تغيير قيم المفتاح الأساسي
- يجب إعطاء المفتاح الأساسي قيمة عند إدراج سجل جديد.

What is Composite Key?

– A composite key is a primary key composed of multiple columns used to identify a record uniquely

ما هو المفتاح المركب؟

– المفتاح المركب هو مفتاح أساسي يتكون من عدة أعمدة تستخدم لتعريف السجل بشكل فريد.

What is a Foreign Key in SQL?

Foreign Key references the primary key of another Table! It helps connect database Tables.

ما هو المفتاح الخارجي فيSQL ؟

يشير المفتاح الخارجي إلى المفتاح الأساسي لجدول آخر! يساعد على ربط جداول قاعدة البيانات.

Foreign key has the following attributes:

- A foreign key can have a different name from its primary key
- It ensures rows in one table have corresponding rows in another
- Unlike the Primary key, they do not have to be unique. Most often they aren't
- Foreign keys can be null even though primary keys can not

يحتوي المفتاح الخارجي على السمات التالية:

- يمكن أن يكون للمفتاح الخارجي اسم مختلف عن مفتاحه الأساسي
- فهو يضمن أن الصفوف في جدول واحد لها صفوف مقابلة في جدول آخر
- وعلى عكس المفتاح الأساسي، ليس من الضروري أن تكون فريدة. في أغلب الأحيان لا يكونون كذلك
- يمكن أن تكون المفاتيح الخارجية فارغة على الرغم من أن المفاتيح الأساسية لا يمكن أن تكون فارغة

What are transitive functional dependencies?

A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change

Consider the table 1. Changing the non-key column Full Name may change Salutation.

ما هي التبعيات الوظيفية متعدية؟

قد تتسبب التبعية الوظيفية المتعدية عند تغيير عمود غير رئيسي في تغيير أي من الأعمدة الأخرى غير الرئيسية

خذ بعين الاعتبار الجدول ١. قد يؤدي تغيير العمود غير الرئيسي "الاسم الكامل" إلى تغيير التحية.

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Change in Name

May Change Salutation

What is SQL Normalization?

- Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update, and Deletion Anomalies.
- Normalization rules divide larger tables into smaller tables and link them using relationships.
- The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

ما هو تطبيع قاعدة البيانات SQL ؟

- التطبيع هو أسلوب تصميم قاعدة بيانات يقلل من تكرار البيانات ويزيل الخصائص غير المرغوب فيها مثل حالات الإدراج والتحديث والحذف الشاذة.
- تقوم قواعد التطبيع بتقسيم الجداول الكبيرة إلى جداول أصغر وربطها باستخدام العلاقات.
- الغرض من التطبيع في SQL هو التخلص من البيانات الزائدة (المتكررة) والتأكد من تخزين البيانات بشكل منطقي.

- Normalization in SQL databases offers several advantages such as data integrity, reduction of data redundancy, efficient data storage, improved query performance (in many cases), easier data maintenance, scalability, enhanced data modeling, consistency, reduction of anomalies, and simplified schema changes.

- يوفر التطبيع في قواعد بيانات SQL العديد من المزايا مثل تكامل البيانات، وتقليل تكرار البيانات، وتخزين البيانات بكفاءة، وتحسين أداء الاستعلام (في العديد من الحالات)، وسهولة صيانة البيانات، وقابلية التوسع، ونمذجة البيانات المحسنة، والاتساق، وتقليل الحالات الشاذة، والمخطط المبسط التغييرات.

- Different levels of normalization exist, such as 1NF (First Normal Form), 2NF (Second Normal Form), 3NF (Third Normal Form), and BCNF (Boyce-Codd Normal Form)

- توجد مستويات مختلفة للتطبيع، مثل 1NF (نموذج التطبيع الاول)، 2NF (نموذج التطبيع الثاني)، 3NF (نموذج التطبيع الثالث) ، وBCNF (نموذج تطبيع بويس كود)

Note: in most practical applications, normalization achieves its best in the 3rd Normal Form.

ملاحظة: في معظم التطبيقات العملية، يحقق التطبيع أفضل ما لديه في النموذج العادي الثالث.

1NF (First Normal Form):

In 1NF, each column in a table contains only atomic values, meaning it cannot be further divided. There should be no repeating groups or arrays of values within a single column. Each row in the table should be uniquely identifiable.

في نموذج التطبيع الاول، يحتوي كل عمود في الجدول على قيم ذرية فقط، مما يعني أنه لا يمكن تقسيمه بشكل أكبر. لا ينبغي أن تكون هناك مجموعات متكررة أو صفوف من القيم داخل عمود واحد. يجب أن يكون كل صف في الجدول قابلاً للتعريف بشكل فريد.

Example:

Original Table:

CustomerID	Name	Phone no.
1	Huda	8978847383
2	Ahmed	7899748899, 8899278299
3	Mohammed	9877382892

1NF Table:

CustomerID	Name	Phone no.
1	Huda	8978847383
2	Ahmed	7899748899
3	Mohammed	9877382892
2	Ahmed	8899278299

2NF (Second Normal Form):

In 2NF, the table is already in 1NF, and each non-key column is dependent on the entire primary key. If there are partial dependencies, those columns should be moved to a separate table.

في نموذج التطبيع الثاني، يكون الجدول موجودًا بالفعل في نموذج التطبيع الاول، ويعتمد كل عمود غير رئيسي على المفتاح الأساسي بالكامل. إذا كانت هناك تبعيات جزئية، فيجب نقل هذه الأعمدة إلى جدول منفصل.

Example:

Original Table:

OrderID	ProductID	ProductName	Category	Price
1	1	Laptop	Electronics	1000
2	2	Smartphone	Electronics	800
3	3	Laptop	Electronics	900

2NF Table:

Table 1: Products

ProductID	ProductName	Category
1	Laptop	Electronics
2	Smartphone	Electronics

Table 2: Orders

OrderID	ProductID	Price
1	1	1000
2	2	800
3	1	900

3NF (Third Normal Form):

In 3NF, the table is already in 2NF, and there are no transitive dependencies. Non-key columns should not depend on other non-key columns. If there are such dependencies, those columns should be moved to a separate table.

في نموذج التطبيع الثالث، يكون الجدول موجودًا بالفعل في نموذج التطبيع الثاني، ولا توجد تبعيات متعدية. لا ينبغي أن تعتمد الأعمدة غير الرئيسية على أعمدة أخرى غير رئيسية. إذا كانت هناك مثل هذه التبعيات، فيجب نقل هذه الأعمدة إلى جدول منفصل.

Example:

Original Table:

CustomerID	OrderID	ProductID	ProductName	Price	CustomerName	CustomerEmail
1	1	1	Laptop	1000	Huda	Huda@example.com
2	2	2	Smartphone	800	Ahmed	Ahmed@example.com
3	3	1		900	Ali	Ali@example.com

3NF Table:

Table 1: Customers

CustomerID	CustomerName	CustomerEmail
1	Huda	Huda@example.com
2	Ahmed	Ahmed@example.com
3	Ali	Ali@example.com

Table 2: Products

ProductID	ProductName
1	Laptop
2	Smartphone

Table 3: Orders

OrderID	CustomerID	ProductID	Price
1	1	1	1000
2	2	2	800
3	3	1	900

BCNF (Boyce-Codd Normal Form):

BCNF is an advanced form of normalization that addresses certain anomalies that can occur in 3NF. It ensures that there are no non-trivial functional dependencies of non-key attributes on a candidate key. Achieving BCNF involves decomposing tables further if necessary.

BCNFهو شكل متقدم من أشكال التطبيع الذي يعالج بعض الشذوذ التي قد تحدث في نموذج التطبيع الثالث. ويضمن عدم وجود تبعيات وظيفية غير طفيفة للسمات غير الرئيسية على مفتاح مرشح. ويتضمن تحقيق BCNFتحليل الجداول بشكل أكبر إذا لزم الأمر.

SQL Database Basics

المرحلة الثانية/ الفصل الدراسي الاول

قسم تقنيات أنظمة الحاسوب

المعهد التقني في البصرة

اعداد

المدرس المساعد زينب محمد جوار

السنة الدراسية: ٢٠٢٤–٢٠٢٥

Introduction to SQL

SQL (**S**tructured **Q**uery **L**anguage) is a computer programming language designed specifically for maintaining and modifying databases in **R**elational **D**atabase **M**anagement **S**ystems (**RDBM**S).. It offers statements and commands for creating database structures. Users can interact with databases to access, modify, and manage structured data by creating queries. Regardless of the underlying database management system, SQL offers a consistent and effective way to work with databases.

SQL (لغة الاستعلامات المنظمة/ المهيكلة) هي لغة برمجة حاسوبية مصممة خصيصًا للحفاظ على قواعد البيانات وتعديلها في أنظمة إدارة قواعد البيانات العلائقية (RDBMS). وهي توفر عبارات وأوامر لإنشاء هياكل قواعد البيانات. ويمكن للمستخدمين التفاعل مع قواعد البيانات للوصول إلى البيانات المنظمة وتعديلها وإدارتها من خلال إنشاء استعلامات. وبغض النظر عن نظام إدارة قواعد البيانات الأساسي، توفر SQL طريقة متسقة وفعالة للعمل مع قواعد البيانات.

What Can SQL do?

SQL can execute queries against a database
SQL can retrieve data from a database
SQL can insert records in a database
SQL can update records in a database
SQL can delete records from a database
SQL can create new databases
SQL can create new tables in a database
SQL can create stored procedures in a database
SQL can create views in a database
SQL can set permissions on tables, procedures, and views

ماذا يمكن أن تفعل SQL ؟
يمكن لـ SQL تنفيذ الاستعلامات على قاعدة البيانات
يمكن لـ SQL استرداد البيانات من قاعدة البيانات
يمكن لـ SQL إدراج السجلات في قاعدة البيانات
يمكن لـ SQL تحديث السجلات في قاعدة البيانات
يمكن لـ SQL حذف السجلات من قاعدة البيانات
يمكن لـ SQL إنشاء قواعد بيانات جديدة
يمكن لـ SQL إنشاء جداول جديدة في قاعدة البيانات
يمكن لـ SQL إنشاء الإجراءات المخزنة في قاعدة البيانات
يمكن لـ SQL إنشاء طرق عرض في قاعدة البيانات
يمكن لـ SQL تعيين الأذونات على الجداول والإجراءات وطرق العرض

SQL constraints

- Rules for the data in a table can be specified using SQL constraints.
- The kinds of data that can be entered into a table are restricted by constraints. This guarantees the reliability and accuracy of the data in the table. The action is stopped if there is a violation between the constraint and the data action.
- Column-level or table-level constraints are both possible. Table level restrictions apply to the entire table, while column level constraints just affect the specified column.

- يمكن تحديد قواعد البيانات في جدول باستخدام قيود.SQL
- يتم تقييد أنواع البيانات التي يمكن إدخالها في جدول بالقيود. وهذا يضمن موثوقية ودقة البيانات في الجدول. يتم إيقاف الإجراء إذا كان هناك انتهاك بين القيد وإجراء البيانات.
- يمكن تطبيق القيود على مستوى العمود أو على مستوى الجدول. تنطبق القيود على مستوى الجدول على الجدول بأكمله، بينما تؤثر القيود على مستوى العمود فقط على العمود المحدد.

In SQL, the following restrictions are frequently applied:
فيSQL ، يتم تطبيق القيود التالية بشكل متكرر:

- **NOT NULL:** A column cannot have a NULL value by using the NOT NULL flag.
  لا يمكن أن يحتوي العمود على قيمة NULL باستخدام علامة.NOT NULL
- **UNIQUE:** A unique value makes sure that each value in a column is distinct.
  تضمن القيمة الفريدة أن تكون كل قيمة في العمود مميزة.
- **PRIMARY KEY:** A NOT NULL and UNIQUE combination. Uniquely identifies each table row.
  مزيج من NOT NULL و .UNIQUEيحدد كل صف في الجدول بشكل فريد.
- **FOREIGN KEY:** Prevent acts that would break linkages between tables.
  يمنع الأفعال التي قد تقطع الارتباطات بين الجداول.
- **CHECK-** Verifies if the values in a column meet a certain requirement.
  يتحقق مما إذا كانت القيم في العمود تلبي متطلبًا معينًا.
- **DEFAULT:** If no value is specified, DEFAULT sets a default value for the column.
  إذا لم يتم تحديد أي قيمة، يقوم DEFAULT بتعيين قيمة افتراضية للعمود.
- **CREATE INDEX -** Used to easily create and access data from the database.
  يستخدم لإنشاء البيانات والوصول إليها بسهولة من قاعدة البيانات.

SQL command types

1. **Data Definition Language (DDL):** The commands that fall under this group provide the ability to define data, its form, and the way it is linked to each other through the use of commands to create tables and a database. Common DDL commands include **CREATE**, **ALTER**, and **DROP**.

أنواع أوامرSQL

١ ـ لغة تعريف البيانات :(DDL) توفر الأوامر التي تندرج ضمن هذه المجموعة إمكانية تعريف البيانات وشكلها وطريقة ربطها ببعضها البعض من خلال استخدام الأوامر لإنشاء الجداول وقاعدة البيانات. تتضمن أوامر DDL الشائعة CREATE و ALTERو.DROP

2. **Data Manipulation Language (DML):** This group contains sentences whose purpose is to give the ability to deal with data without affecting its structure and general form so that you can query data, add records, and delete or modify them. Common DML commands include **SELECT**, **INSERT**, **UPDATE**, and **DELETE**.

٢ ـ لغة معالجة البيانات :(DML) تحتوي هذه المجموعة على جمل هدفها إعطاء القدرة على التعامل مع البيانات دون التأثير على بنيتها وشكلها العام بحيث يمكنك الاستعلام عن البيانات وإضافة السجلات وحذفها أو تعديلها. تتضمن أوامر DML الشائعة SELECT و INSERTو UPDATE و.DELETE

3. **Data Control Language (DCL):** This set of commands helps in defining the permissions that can be granted or taken away from users in the database. Common DCL commands include **GRANT** and **REVOKE**, which are used to assign or remove permissions to users or roles.

٣ ـ لغة التحكم في البيانات :(DCL) تساعد مجموعة الأوامر هذه في تحديد الأذونات التي يمكن منحها أو أخذها من المستخدمين في قاعدة البيانات. تتضمن أوامر DCL الشائعة GRANT وREVOKE، والتي يتم استخدامها لتعيين أو إزالة الأذونات للمستخدمين أو الأدوار.

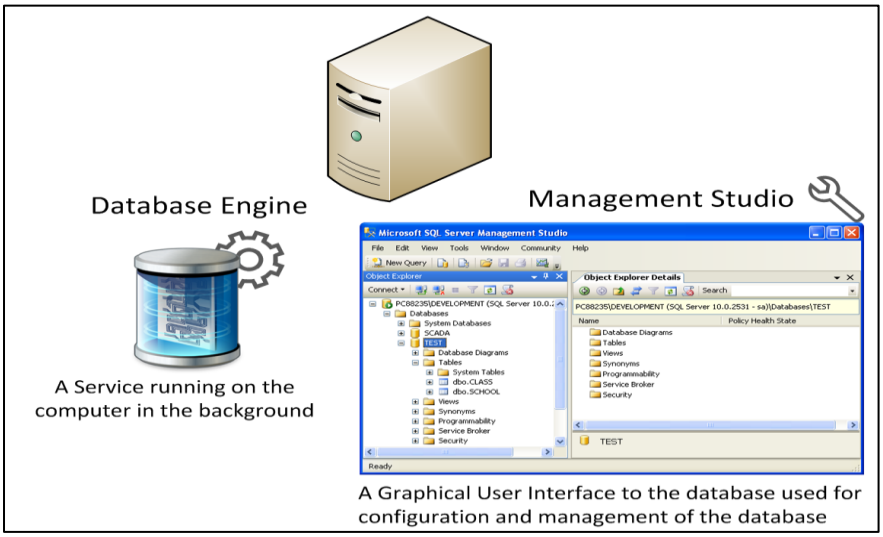
.

Introduction to SQL Server مقدمة إلى خادم SQL

- Microsoft is the vendor of SQL Server. We have different editions of SQL Server, where SQL Server Express is free to download and use.
- SQL Server consists of a Database Engine and a Management Studio (and lots of other stuff that we will not mention here). The Database engine has no graphical interface - it is just a service running in the background of your computer (preferably on the server). The Management Studio is a graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both).

Microsoft هي بائع SQL Server. لدينا إصدارات مختلفة منSQL Server ، حيث يكون SQL Server Expressمجانيًا للتنزيل والاستخدام.

تكون SQL Server من محرك قاعدة البيانات واستوديو الإدارة (والكثير من الأشياء الأخرى التي لن نذكرها هنا). لا يحتوي محرك قاعدة البيانات على واجهة رسومية - فهو مجرد خدمة تعمل في خلفية جهاز الكمبيوتر الخاص بك (يفضل أن يكون ذلك على الخادم). يعد Management Studioأداة رسومية لتكوين وعرض المعلومات في قاعدة البيانات. يمكن تثبيته على الخادم أو على العميل (أو كليهما).



SQL Server

SQL Server Management Studio

- **S**QL **S**erver **M**anagement **S**tudio (**SSMS**) is a GUI tool included with SQL Server for configuring, managing, and administering all components within Microsoft SQL Server.

ستوديو إدارة خادم SQL

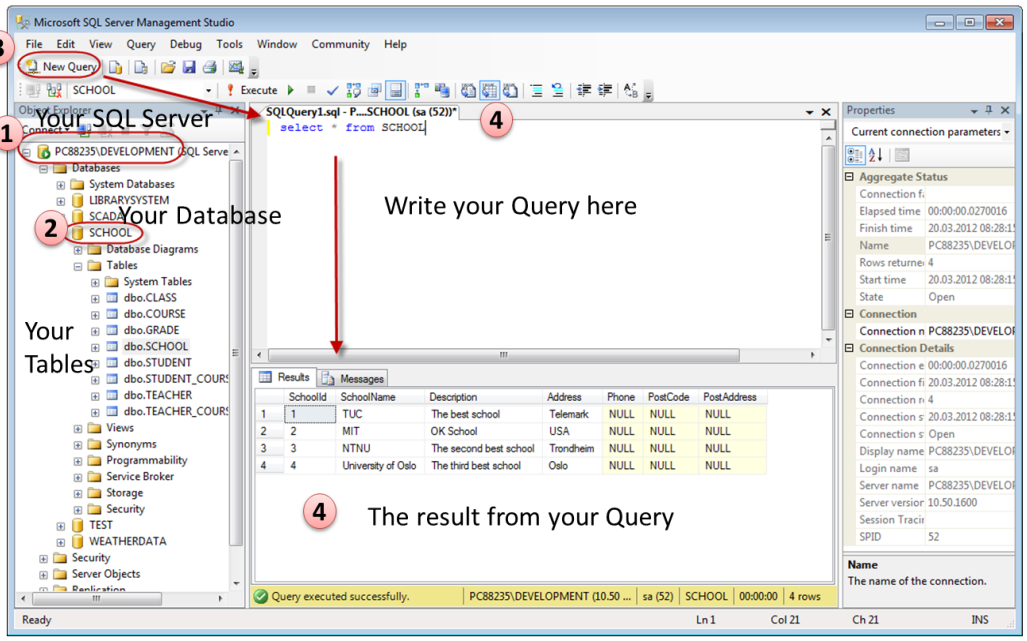
- يعد SQL Server Management Studio أداة واجهة المستخدم الرسومية المضمنة في SQL Serverلتكوين جميع المكونات وإدارتها داخل.Microsoft SQL Server

- The tool includes both script editors and graphical tools that work with objects and features of the server.

- تشتمل الأداة على محررات البرامج النصية والأدوات الرسومية التي تعمل مع كائنات وميزات الخادم.

- A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

- الميزة المركزية لـ SQL Server Management Studio هيObject Explorer ، والتي تسمح للمستخدم باستعراض أي من الكائنات الموجودة داخل الخادم وتحديدها والتصرف بناءً عليها. ويمكن استخدامه لمراقبة خطط الاستعلام وتحليلها بشكل مرئي وتحسين أداء قاعدة البيانات، من بين أشياء أخرى. يمكن أيضًا استخدام SQL Server Management Studio لإنشاء قاعدة بيانات جديدة، أو تغيير أي مخطط قاعدة بيانات موجود عن طريق إضافة أو تعديل الجداول والفهارس، أو تحليل الأداء. ويتضمن نوافذ الاستعلام التي توفر واجهة مستخدم رسومية لكتابة الاستعلامات وتنفيذها.



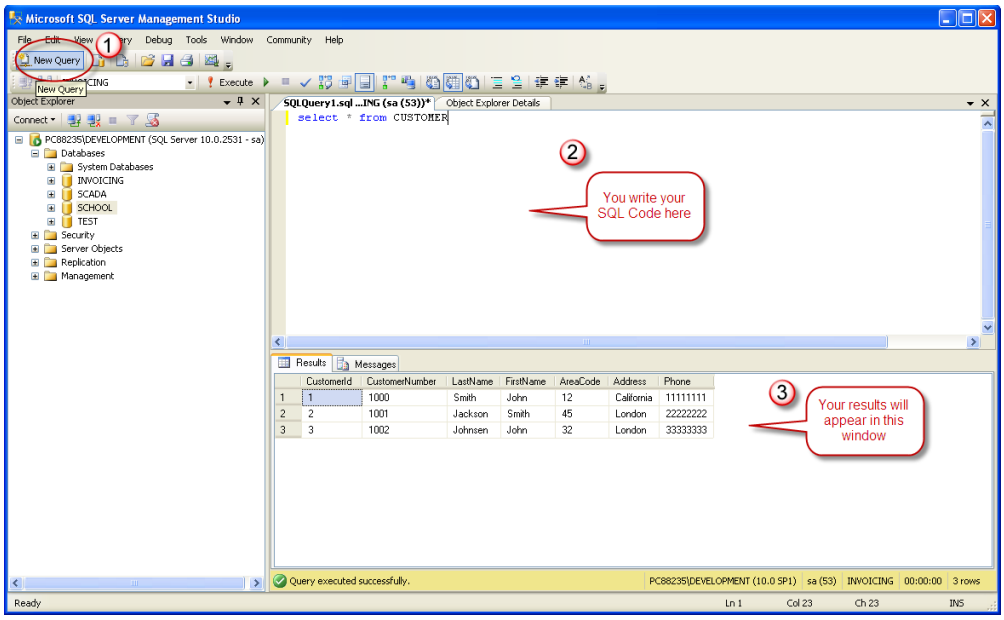
SQL Server Management Studio

- When creating SQL commands and queries, the “Query Editor” (select “New Query” from the Toolbar) is used (shown in the figure above).

- عند إنشاء أوامر واستعلاماتSQL ، يتم استخدام "محرر الاستعلام" (اختر "استعلام جديد" من شريط الأدوات) (كما هو موضح في الشكل أعلاه).

- With SQL and the “Query Editor” we can do almost everything with code, but sometimes it is also a good idea to use the different Designer tools in SQL to help us do the work without coding (so much).

- باستخدام SQL و"محرر الاستعلام" يمكننا القيام بكل شيء تقريبًا باستخدام التعليمات البرمجية، ولكن في بعض الأحيان يكون من الجيد أيضًا استخدام أدوات المصمم المختلفة في SQL لمساعدتنا في تنفيذ العمل بدون تعليمات برمجية (كثيرًا).

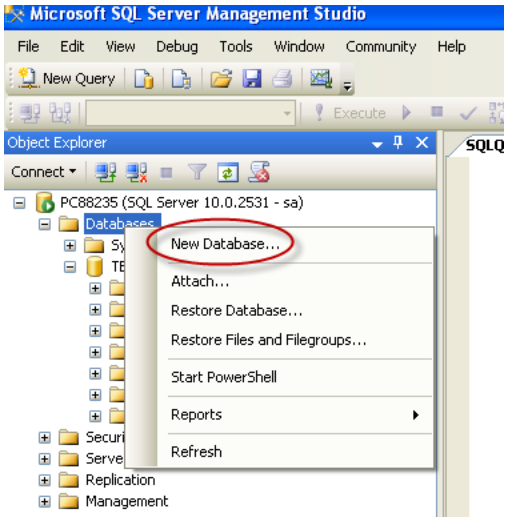


Create a new database using the wizard

It is quite simple to create a new database in Microsoft SQL Server using the wizard. Just right-click on the “Databases” node and select “New Database…”

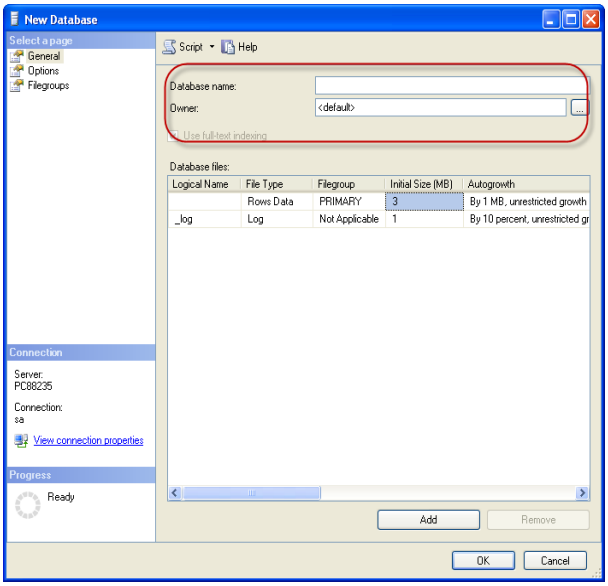
إنشاء قاعدة بيانات جديدة باستخدام المعالج

من السهل جدًا إنشاء قاعدة بيانات جديدة في Microsoft SQL Server. فقط انقر بزر الماوس الأيمن على عقدة "قواعد البيانات" وحدد "قاعدة بيانات جديدة"...



There are lots of settings you may set regarding your database, but the only information you must fill in is the name of your database:

هناك الكثير من الإعدادات التي يمكنك تعيينها فيما يتعلق بقاعدة البيانات الخاصة بك، ولكن المعلومات الوحيدة التي يجب عليك ملؤها هي اسم قاعدة البيانات الخاصة بك :



You may also use the SQL language to create a new database, but sometimes it is easier to just use the built-in features in the Management Studio.

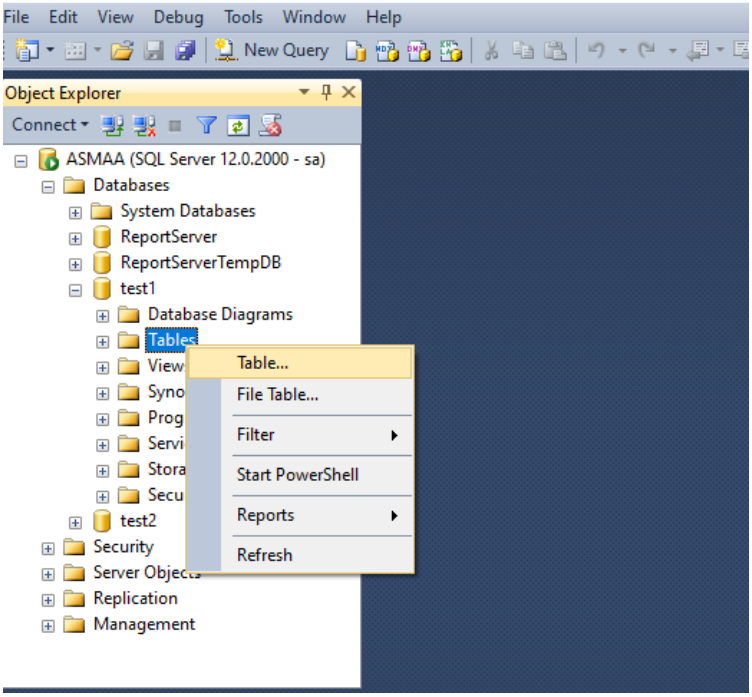
يمكنك أيضًا استخدام لغة SQL لإنشاء قاعدة بيانات جديدة، ولكن في بعض الأحيان يكون من الأسهل استخدام الميزات المضمنة في Management Studio فقط.

Create a table using the wizard

Open the database that you created, right-click on the selection Tables, a menu appears, choose a “Table….” from it.

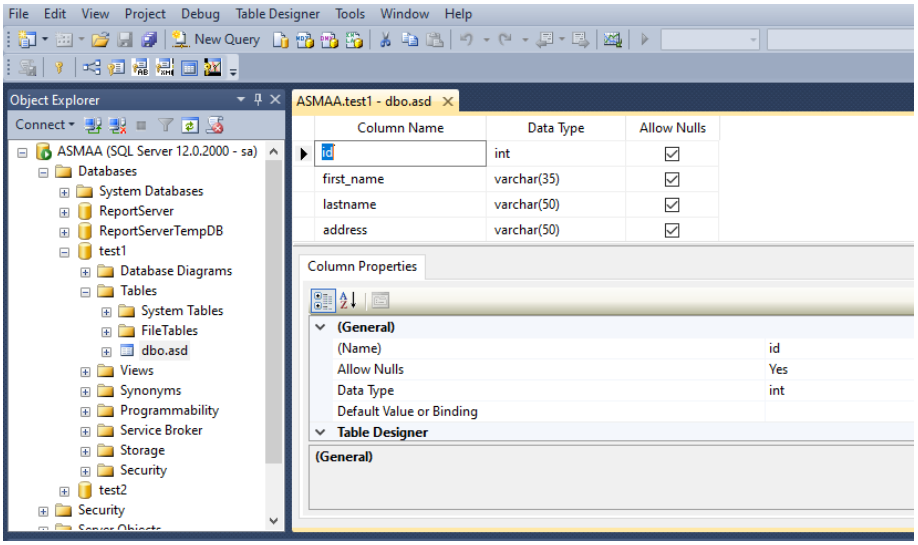
إنشاء جدول باستخدام المعالج

افتح قاعدة البيانات التي قمت بإنشائها، وانقر بزر الماوس الأيمن على تحديد الجداول، وستظهر قائمة، اختر "جدول...". منه.



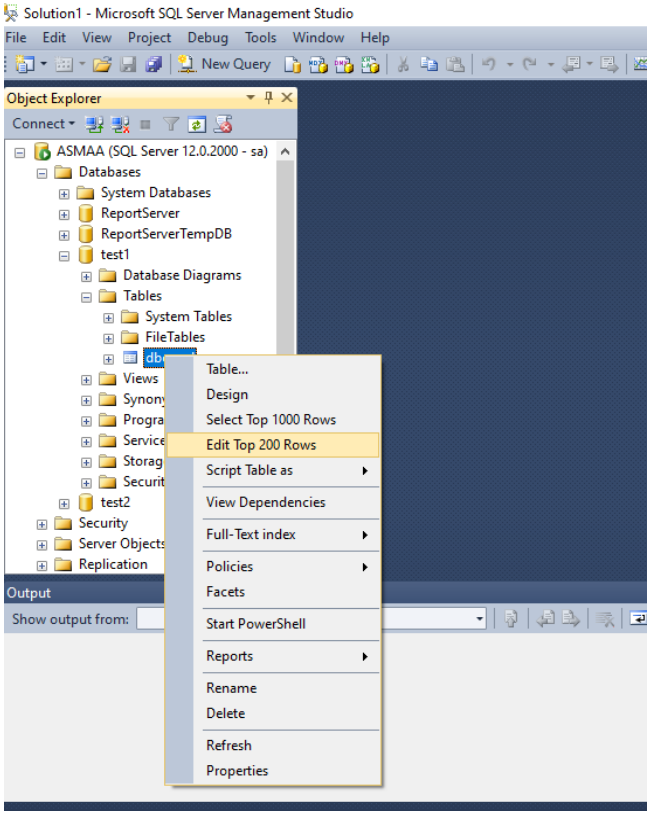
After that, the table design interface appears, which contains the column name and the graph type, add columns and select the graphic type for each column.

بعد ذلك تظهر واجهة تصميم الجدول والتي تحتوي على اسم العمود ونوع الرسم البياني وإضافة الأعمدة واختيار نوع الرسم لكل عمود.



To add data to the table, right-click on the table name, after saving it, a menu appears, choose from it “Edit Top 200 Rows”.

لإضافة بيانات إلى الجدول، انقر بزر الماوس الأيمن على اسم الجدول، وبعد حفظه تظهر قائمة، اختر منها "Edit Top 200 Rows" .



SQL Database Basics

المرحلة الثانية/ الفصل الدراسي الاول

قسم تقنيات أنظمة الحاسوب

المعهد التقني في البصرة

اعداد

المدرس المساعد زينب محمد جوار

السنة الدراسية: ٢٠٢٤–٢٠٢٥

SQL DDL commands using Scripts

أوامر SQL DDL باستخدام الايعازات

The SQL CREATE DATABASE Statement

The CREATE DATABASE statement is used to create a new SQL database.

امر لغة SQL لإنشاء قاعدة بيانات

يتم استخدام امر CREATE DATABASE لإنشاء قاعدة بيانات SQL جديدة.

Syntax صيغة الامر

```
CREATE DATABASE databasename;
```

SQL CREATE DATABASE Example

The following SQL statement creates a database called "testDB":

مثال امر لغة SQL لإنشاء قاعدة بيانات

يقوم امر SQL التالية بإنشاء قاعدة بيانات تسمى:"testDB"

Example مثال

```
CREATE DATABASE testDB;
```

The SQL DROP DATABASE Statement

The DROP DATABASE statement is used to drop an existing SQL database.

امر لغة SQL حذف (اسقاط) قاعدة بيانات

يتم استخدام امر DROP DATABASE لحذف قاعدة بيانات SQL موجودة.

Syntax صيغة الامر

```
DROP DATABASE databasename;
```

SQL DROP DATABASE Example

The following SQL statement drops the existing database "testDB":

مثال امر لغة SQL لحذف (اسقاط) قاعدة بيانات

يقوم امر SQL التالية بحذف قاعدة بيانات الموجودة:"testDB"

Example مثال

```
DROP DATABASE testDB;
```

The SQL CREATE TABLE Statement

The CREATE TABLE statement is used to create a new table in a database.

امر لغة SQL لأنشاء جدول

يتم استخدام امر CREATE TABLE لإنشاء جدول جديد في قاعدة البيانات.

Syntax صيغة الامر

```
CREATE TABLE table_name (
  column1 datatype,
  column2 datatype,
  column3 datatype,
  ....
);
```

The column parameters specify the names of the columns of the table.

تحدد معلمات column أسماء أعمدة الجدول.

The datatype parameter specifies the type of data the column can hold.

تحدد المعلمة datatype نوع البيانات التي يمكن أن يحتوي عليها العمود.

The data type can be one of the following categories:

- **Numeric** such as INT, TINYINT, BIGINT, FLOAT, REAL, etc.
- **Date and Time** such as DATE, TIME, DATETIME, etc.
- **Character and String** data types such as CHAR, VARCHAR, TEXT, etc.
- **Unicode character string** such as NCHAR, NVARCHAR, NTEXT, etc.
- **Binary** such as BINARY, VARBINARY, etc.
- **Other** such as CLOB, BLOB, XML, CURSOR, TABLE, etc.

يمكن أن يكون نوع البيانات إحدى الفئات التالية:

- رقمية مثلINT ، وTINYINT، وBIGINT، وFLOAT، وREAL، وما إلى ذلك.
- التاريخ والوقت مثلDATE ، TIME، DATETIME، إلخ.
- الأحرف والسلاسل مثل CHAR و VARCHARو TEXTوما إلى ذلك.
- سلسلة أحرف Unicode مثلNCHAR ، وNVARCHAR، وNTEXT، وما إلى ذلك.
- ثنائي مثل BINARY و VARBINARY وما إلى ذلك.
- أخرى مثل CLOB و BLOBو XMLو CURSORو TABLEوما إلى ذلك.

SQL CREATE TABLE Examples

امثلة امر لغة SQL لأنشاء جدول

Example-1 مثال ١

The following example creates a table called "Persons" that contains five columns: PersonID, LastName, FirstName, Address, and City:

يقوم المثال التالي بإنشاء جدول يسمى "الأشخاص" يحتوي على خمسة أعمدة: معرف الشخص، واسم العائلة، والاسم الأول، والعنوان، والمدينة:

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

The PersonID column is of type int and will hold an integer.

عمود PersonID من النوع int وسيحتوي على عدد صحيح.

The LastName, FirstName, Address, and City columns are of type varchar and will hold characters, and the maximum length for these fields is 255 characters.

أعمدة اسم العائلة والاسم الأول والعنوان والمدينة هي من النوع varchar وستحتوي على أحرف، والحد الأقصى لطول هذه الحقول هو ٢٥٥ حرفًا.

The empty "Persons" table will now look like this:

سيبدو جدول "الأشخاص" الفارغ الآن كما يلي:

PersonID	LastName	FirstName	Address	City

Example-2 مثال ٢

The following example creates a table called " Employee" that contains five columns: EmpId, LastName, FirstName, Address, and City:

يقوم المثال التالي بإنشاء جدول يسمى "الموظف" يحتوي على خمسة أعمدة :معرف الموظف، واسم العائلة، والاسم الأول، والعنوان، والمدينة:

```
CREATE TABLE Employee (
    EmpId int,
    LastName varchar(25),
    FirstName varchar(25),
    Address varchar(100),
    City varchar(20)
);
```

The EmpId column is of type int and will hold an integer.

العمود EmpId من النوع int وسيحتوي على عدد صحيح.

The LastName, FirstName, Address, and City columns are of type varchar and will hold characters and the maximum length for these fields is 255 characters.

أعمدة اسم العائلة والاسم الأول والعنوان والمدينة هي من النوع varchar وستحتوي على أحرف والحد الأقصى لطول هذه الحقول هو ٢٥٥ حرفًا.

The empty " Employee" table will now look like this:

سيبدو جدول "الموظف" الفارغ الآن كما يلي:

EmpId	LastName	FirstName	Address	City

The SQL INSERT INTO Statement

The INSERT INTO statement is used to insert new records in a table.

It is possible to write the INSERT INTO statement in two ways.

امر لغة SQL لادراج قيم في الجدول

يتم استخدام امر INSERT INTO لإدراج سجلات جديدة في الجدول.

من الممكن كتابة عبارة INSERT INTO بطريقتين.

Syntax صيغة الامر

The first way specifies both the column names and the values to be inserted.

If you are adding values for all the columns of the table, then no need to specify the column names in the SQL query. However, make sure that the order of the values is in the same order as the columns in the table.

تحدد الطريقة الأولى أسماء الأعمدة والقيم التي سيتم إدراجها.

إذا كنت تقوم بإضافة قيم لجميع أعمدة الجدول، فلا داعي لتحديد أسماء الأعمدة في استعلام SQL. ومع ذلك، تأكد من أن ترتيب القيم بنفس ترتيب الأعمدة في الجدول.

First way: الطريقة الاولى

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

Second way: الطريقة الثانية

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

SQL INSERT INTO Example

مثال امر لغة SQL لادراج قيم في جدول

Insert value in the first way. The column names are used here

أدخل القيمة بالطريقة الأولى. يتم استخدام أسماء الأعمدة هنا

```
INSERT INTO Employee (EmpId,LastName,FirstName,ADDRESS,City)
VALUES (1, 'Ali', 'Wael', 'Iraq', 'Basra' );
INSERT INTO Employee (EmpId,LastName,FirstName,ADDRESS,City)
VALUES (2, 'Mohammed', 'Saad', 'Iraq', 'Baghdad' );
```

Insert value in the second way.

أدخل القيمة بالطريقة الثانية.

```
INSERT INTO Employee
VALUES (3, 'Huda', 'Salim', 'Iraq', 'Basra' );
```

The SQL SELECT Statement

امر لغة SQL تحديد قيم في جدول

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

يتم استخدام امر SELECT لتحديد البيانات من قاعدة البيانات.

يتم تخزين البيانات التي تم إرجاعها في جدول نتائج يسمى مجموعة النتائج.

Syntax

```
SELECT column1, column2, ...
FROM table_name;
```

Here, column1, column2, ... are the field names of the table you want to select from the data. If you want to select all the fields available in the table, use the following syntax:

هنا، column1، column2، ...هي أسماء حقول الجدول الذي تريد تحديده من البيانات. إذا كنت تريد تحديد كافة الحقول المتوفرة في الجدول، استخدم بناء الجملة التالي:

```
SELECT * FROM table_name;
```

If the above query is executed, then all record is displayed.

إذا تم تنفيذ الاستعلام أعلاه، فسيتم عرض كل السجلات.

Example مثال

```
SELECT EmpId, LastName FROM Employee;

SELECT * FROM Employee;
```

The SQL DROP TABLE Statement

امر لغة SQL حذف (اسقاط) جدول

The DROP TABLE statement is used to drop an existing table in a database.

يتم استخدام امر DROP TABLE لإسقاط جدول موجود في قاعدة البيانات.

Syntax صيغة الامر

```
DROP TABLE table_name;
```

SQL DROP TABLE Example

مثال امر لغة SQL لحذف (اسقاط) جدول

The following SQL statement drops the existing table "Persons":

تقوم امر SQL التالية بإسقاط الجدول الموجود "الأشخاص":

Example مثال

```
DROP TABLE Persons;
```

SQL TRUNCATE TABLE

The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

امر لغة SQL لحذف (اقتطاع) قيم بيانات جدول

يتم استخدام عبارة TRUNCATE TABLE لحذف البيانات الموجودة داخل الجدول، وليس الجدول نفسه.

Syntax صيغة الامر

```
TRUNCATE TABLE table_name;
```

The SQL ALTER TABLE Statement

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

امر لغة SQL لتغيير جدول

يتم استخدام عبارة ALTER TABLE لإضافة أو حذف أو تعديل الأعمدة في جدول موجود.

يتم استخدام عبارة ALTER TABLE أيضًا لإضافة وإسقاط شروط مختلفة على جدول موجود.

ALTER TABLE - ADD Column

To add a column in a table, use the following syntax:

تغيير الجدول ـ إضافة عمود

لإضافة عمود في جدول، استخدم صيغة الامر التالي:

Syntax صيغة الامر

```
ALTER TABLE table_name
ADD column_name datatype;
```

ALTER TABLE - ADD Column Example

The following SQL statement adds an "Email" column to the "Employee" table:

يضيف امر SQL التالية عمود "البريد الإلكتروني" إلى جدول "الموظف":

مثال لتغيير جدول ـ اضافة عمود

Example مثال

```
ALTER TABLE Employee
ADD Email varchar(255);
```

ALTER TABLE - DROP COLUMN

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

تغيير الجدول - حذف (اسقاط) عمود

لحذف عمود في جدول، استخدم الصيغة التالية (لاحظ أن بعض أنظمة قواعد البيانات لا تسمح بحذف عمود):

Syntax

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

ALTER TABLE - DROP COLUMN Example

مثال لتغيير جدول - حذف (اسقاط) عمود

The following SQL query deletes the "Email" column from the "Employee" table:

يقوم استعلام SQL التالي بحذف عمود "البريد الإلكتروني" من جدول "الموظف":

Example

```
ALTER TABLE Employee
DROP COLUMN Email;
```

SQL Database Basics

المرحلة الثانية/ الفصل الدراسي الاول

قسم تقنيات أنظمة الحاسوب

المعهد التقني في البصرة

اعداد

المدرس المساعد زينب محمد جوار

السنة الدراسية: ٢٠٢٤–٢٠٢٥

ALTER TABLE - ALTER/MODIFY DATATYPE

تغيير الجدول ـ تغيير/تعديل نوع البيانات

To change the datatype of a column in a table, use the following syntax:

لتغيير نوع بيانات عمود في جدول، استخدم الصيغة التالية:

Syntax

```
ALTER TABLE table_name
ALTER COLUMN column_name datatype;
```

ALTER TABLE - ALTER/MODIFY DATATYPE Example

مثال لتغيير جدول ـ تغيير/تعديل نوع البيانات

The following SQL query changes the datatype of a column "*LastName*" of the datatype "varchar(25)" in a table "Employee" to be in a new datatype of "*nchar(10)*" :

يقوم استعلام SQL التالي بتغيير نوع بيانات العمود "*LastName*" ذي النوع البياني"varchar(25)" في الجدول "Employee" الى النوع البياني الجديد "*nchar(10)*"":

Example

```
ALTER TABLE Employee
ALTER COLUMN LastName nchar(10);
```

Note: To change the datatype of any column in a table, the new datatype must be compatible with the entered data.

ملاحظة: لتغيير نوع بيانات أي عمود في الجدول، يجب أن يكون نوع البيانات الجديد متوافقاً مع البيانات المدخلة.

The SQL SELECT DISTINCT Statement

امر لغة SQL لإرجاع القيم الفريدة

The SELECT DISTINCT statement is used to return only distinct (different) values.

يتم استخدام امر SELECT لإرجاع القيم الفريدة فقط.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

داخل الجدول، غالبًا ما يحتوي العمود على العديد من القيم المكررة؛ وأحيانًا تريد فقط سرد قيم فريدة منها.

Syntax

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

Example

Customers table:

CustomerID	CustomerName	Address	City
1	Ahmed Mohammed	Iraq	Basra
2	Hana Saad	Syria	Damascus
3	Amer Sultan	Iraq	Baghdad
4	Abdulrahman Saleh	Saudi Arabia	Makkah
5	Zainab Mohammed	Iraq	Basra
6	Noof Saud	Saudi Arabia	Jeddah

Select all the different addresses from the "Customers" table:

ارجاع جميع العناوين المختلفة من جدول "Customers".

```
SELECT DISTINCT Address FROM Customers;
```

Result

Address
Iraq
Syria
Saudi Arabia

If you omit the DISTINCT keyword, the SQL statement returns the "Address" value from all the records of the "Customers" table:
إذا قمت بحذف الكلمة الأساسيةDISTINCT ، سوف يعيد امر SQL قيمة "Address" من كافة سجلات جدول "Customers":

```
SELECT Address FROM Customers;
```

Result

Address
Iraq
Syria
Iraq
Saudi Arabia
Iraq
Saudi Arabia

The SQL WHERE Clause

جملة WHERE في لغة SQL

The WHERE clause is used to filter records.

يتم استخدام جملة WHERE لتصفية السجلات (السجلات التي تنطبق عليها شروط معينة ملحقة بجملة WHERE).

It is used to extract only those records that fulfill a specified condition.

يتم استخدامه لاستخراج السجلات التي تفي بشرط محدد فقط.

Syntax

```
SELECT column1, column2, ...
FROM table_name;
WHERE condition;
```

Example

Select all customers from Iraq:

```
SELECT * FROM Customers
WHERE Address='Iraq';
```

Result

CustomerID	CustomerName	Address	City
1	Ahmed Mohammed	Iraq	Basra
3	Amer Sultan	Iraq	Baghdad
5	Zainab Mohammed	Iraq	Basra

Text Fields vs. Numeric Fields

SQL requires **single** quotes around **text** values (most database systems will also allow **double** quotes).

يتطلب SQL علامات اقتباس **مفردة** حول القيم **النصية** (ستسمح معظم أنظمة قواعد البيانات أيضًا بعلامات الاقتباس **المزدوجة**).

However, **numeric** fields should **not** be enclosed in quotes:

ومع ذلك، **لا** ينبغي وضع الحقول **الرقمية** بين علامتي اقتباس:

Example

```
SELECT * FROM Customers
WHERE CustomerID =1;
```

Operators in The WHERE Clause

المعاملات المستخدمة في جملة WHERE

The following operators can be used in the **WHERE** clause:

يمكن استخدام المعاملات التالية في جملة **WHERE**:

Operator المعامل	**Description** **الوصف**	**Example** **مثال**
=	Equal متساوي	SELECT * FROM Customers WHERE CustomerID = 3;
>	Greater than أكبر من	SELECT * FROM Customers WHERE CustomerID > 3;
<	Less than أصغر **من**	SELECT * FROM Customers WHERE CustomerID < 3;
>=	Greater than or equal أكبر من او يساوي	SELECT * FROM Customers WHERE CustomerID >= 3;
<=	Less than or equal أصغر من او يساوي	SELECT * FROM Customers WHERE CustomerID <= 3;
<>	Not equal. Note: In some versions of SQL this operator may be written as !=	SELECT * FROM Customers WHERE CustomerID <> 3;

	غير متساوي. ملاحظة: في بعض إصدارات SQL قد تتم كتابة هذا العامل كـ=!	
BETWEEN	Between a certain range بين نطاق معين	SELECT * FROM Customers WHERE CustomerID BETWEEN 3 AND 6;
IN	To specify multiple possible values for a column لتحديد قيم متعددة محتملة لعمود	SELECT * FROM Customers WHERE City IN ('Basra', 'Baghdad');
LIKE	Search for a pattern (The percent sign (%) represents zero, one, or multiple characters, and the underscore sign (_) represents one, single character) ابحث عن نمط (علامة النسبة المئوية (%) تمثل صفرًا أو حرفًا واحدًا أو عدة أحرف، وعلامة الشرطة السفلية (_) تمثل حرفًا واحدًا)	Ex: Find any address that starts with "S"? SELECT * FROM Customers WHERE Address LIKE 'S%';
		Ex: Find any address that ends with "a"? SELECT * FROM Customers WHERE Address LIKE '%a';
		Ex: Find any address that have "ra" in any position ? SELECT * FROM Customers WHERE Address LIKE '%ra%';
		Ex: Find any address that have "r" in the second position ? SELECT * FROM Customers WHERE Address LIKE '_r%';
		Ex: Find any address that starts with "l" and is at least 2 characters in length? SELECT * FROM Customers WHERE Address LIKE 'l_%';
		Ex: Find any address that starts with "l" and is at least 3 characters in length? SELECT * FROM Customers WHERE Address LIKE 'l__%';
		Ex: Find any address that starts with "l" and ends with "q"? SELECT * FROM Customers WHERE Address LIKE 'l%q';

Note: The WHERE clause is not only used in SELECT statements, it is also used in UPDATE, DELETE, etc.!

Important points نقاط مهم

- When creating a new database successfully, you can view the newly created database in the Object Explorer. If the new database does not appear, you can click the **Refresh** button or press **F5** keyboard to update the object list.

عند إنشاء قاعدة بيانات جديدة بنجاح، يمكنك عرض قاعدة البيانات التي تم إنشاؤها حديثًا في "Object Explorer". إذا لم تظهر قاعدة البيانات الجديدة، فيمكنك النقر فوق الزر **"تحديث"** أو الضغط على لوحة المفاتيح **F5** لتحديث قائمة الكائنات.

- Be Careful before dropping a database. Deleting a database will result in the loss of complete information stored in the database.

كن حذرًا قبل حذف (إسقاط) قاعدة البيانات. سيؤدي حذف قاعدة البيانات إلى فقدان المعلومات الكاملة المخزنة في قاعدة البيانات.

- Tables are database objects that contain all the data in a database (**unit of storage**).

الجداول هي كائنات قاعدة البيانات التي تحتوي على كافة البيانات الموجودة في قاعدة البيانات (**وحدة التخزين**).

- In tables, data is logically organized in a **row-and-column** format similar to a spreadsheet. Each row represents a **unique record**, and each column represents a **field in the record**.

في الجداول، يتم تنظيم البيانات بشكل منطقي بتنسيق **صف وعمود** مشابه لجدول البيانات. يمثل كل صف **سجلاً فريدًا**، ويمثل كل عمود **حقلاً في السجل**.

- The **INSERT INTO** statement is used to insert **new records** in a table, whereas the **ALTER TABLE - ADD** statement is used to add a **new column** in a table.

يتم استخدام عبارة **INSERT INTO** لإدراج **سجلات جديدة** في جدول، بينما يتم استخدام عبارة **ALTER TABLE - ADD** لإضافة **عمود جديد** في الجدول.

- SQL keywords are **not** case sensitive: **select** is the same as **SELECT**.

الكلمات الأساسية لـ SQL **ليست** حساسة لحالة الأحرف: **select** هو نفس **SELECT**

- Semicolon after SQL Statements: this is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.

الفاصلة المنقوطة بعد عبارات SQL : هذه هي الطريقة القياسية لفصل كل عبارة SQL في أنظمة قواعد البيانات التي تسمح بتنفيذ أكثر من امر SQL في نفس الاتصال بالخادم.

SQL Database Basics

المرحلة الثانية/ الفصل الدراسي الاول

قسم تقنيات أنظمة الحاسوب

المعهد التقني في البصرة

اعداد

المدرس المساعد زينب محمد جوار

السنة الدراسية: ٢٠٢٤–٢٠٢٥

The SQL SELECT INTO Statement

The SELECT INTO statement copies data from one table into a new table.

يقوم امر SELECT INTO بنسخ البيانات من جدول واحد إلى جدول جديد.

Syntax1 (Copy all columns into a new table)

Syntax1 (نسخ كافة الأعمدة إلى جدول جديد)

```
SELECT *
INTO new_table
FROM old_table
WHERE condition ;
```

Syntax2 (Copy only some columns into a new table)

Syntax2 (نسخ بعض الأعمدة فقط إلى جدول جديد)

```
SELECT column1, column2, ...
INTO new_table
FROM old_table
WHERE condition ;
```

Note: The new table will be created with the column-names and types as defined in the old table. You can create new column names using the AS clause.

ملاحظة: يتم إنشاء الجدول الجديد بأسماء الأعمدة وأنواعها كما هو محدد في الجدول القديم. يمكنك إنشاء أسماء أعمدة جديدة باستخدام جملة AS .

Example

If we have the **Customers** table in the database **testDB:**

إذا كان لدينا جدول **العملاء** في قاعدة البيانات **testDB**:

CustomerID	CustomerName	Country	City
1	Ahmed Mohammed	Iraq	Basra
2	Hana Saad	Syria	Damascus
3	Amer Sultan	Iraq	Baghdad
4	Abdulrahman Saleh	Saudi Arabia	Makkah
5	Zainab Mohammed	Iraq	Basra
6	Noof Saud	Saudi Arabia	Jeddah

A) The following SQL statement creates a backup copy of table "Customers":

أ - يقوم امر SQL التالي بإنشاء نسخة احتياطية من جدول "العملاء":

```
SELECT * INTO CustomersBackup2023
FROM Customers;
```

B) The following SQL is used to copy the table "Customers" into a new table "CustomersBackup2023" in another database " test2db":

ب - يستخدم امر SQL التالي لنسخ جدول "العملاء" إلى جدول جديد "CustomersBackup2023" في قاعدة بيانات أخرى "test2db":

```
SELECT * INTO test2db.dbo.CustomersBackup2023
FROM testDB.dbo.Customers;
```

C) The following SQL copies only a few columns into a new table:

ت - يقوم امر SQL التالي بنسخ بضعة أعمدة فقط في جدول جديد:

```
SELECT CustomerName, Country INTO CustomersBackup2023
FROM Customers;
```

D) The following SQL copies only the Iraqi customers into a new table:

ث - يقوم امر SQL التالي بنسخ العملاء العراقيين فقط الى جدول جديد:

```
SELECT * INTO CustomersBackup2023
FROM Customers
WHERE Country ='Iraq' ;
```

Note: SELECT INTO can also be used to create a new, empty table using the schema of another. Just add a WHERE clause that causes the query to return no data.

ملاحظة: يمكن أيضًا استخدام SELECT INTO لإنشاء جدول جديد وفارغ باستخدام مخطط جدول آخر. ما عليك سوى إضافة جملة WHERE التي تؤدي إلى عدم إرجاع الاستعلام لأي بيانات.

```
SELECT * INTO new_table
FROM old_table
WHERE 1 = 0 ;
```

The SQL ORDER BY Statement

The ORDER BY statement is used to sort the result-set in ascending or descending order.

The ORDER BY statement sorts the records in ascending order by default or using the **ASC** keyword. To sort the records in descending order, we use the **DESC** keyword.

يتم استخدام امر ORDER BY لفرز مجموعة النتائج بترتيب تصاعدي أو تنازلي.

يقوم امر ORDER BY بفرز السجلات بترتيب تصاعدي بشكل افتراضي او باستخدام ال كلمة الأساسية ASC. لفرز السجلات بترتيب تنازلي، نستخدم الكلمة الأساسية DESC .

Syntax

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

Examples

A) The following SQL statement selects all customers from the "Customers" table, sorted by the "Country" column (in ascending order by default because we did not specify the order type)

أ - يحدد امر SQL التالي جميع العملاء من جدول "العملاء" مرتبة حسب عمود "البلد" (ترتيبا تصاعديا افتراضيًا لأننا لم نحدد نوع الطلب):

```
SELECT * FROM Customers
ORDER BY Country;
```

Result

CustomerID	CustomerName	Country	City
1	Ahmed Mohammed	Iraq	Basra
3	Amer Sultan	Iraq	Baghdad
5	Zainab Mohammed	Iraq	Basra
6	Noof Saud	Saudi Arabia	Jeddah
4	Abdulrahman Saleh	Saudi Arabia	Makkah
2	Hana Saad	Syria	Damascus

B) The following SQL statement selects all customers from the "Customers" table, sorted DESCENDING by the "Country" column:

ب - يحدد امر SQL التالي جميع العملاء من جدول "العملاء"، مرتبة تنازليًا حسب عمود "البلد":

```
SELECT * FROM Customers
ORDER BY Country DESC;
```

Result

CustomerID	CustomerName	Country	City
2	Hana Saad	Syria	Damascus
4	Abdulrahman Saleh	Saudi Arabia	Makkah
6	Noof Saud	Saudi Arabia	Jeddah
1	Ahmed Mohammed	Iraq	Basra
5	Zainab Mohammed	Iraq	Basra
3	Amer Sultan	Iraq	Baghdad

C) The following SQL statement selects all customers from the "Customers" table, sorted (in ascending by default) by the "Country" and the "CustomerName" columns:

ت – يقوم امر SQL التالي بتحديد كافة العملاء من جدول "العملاء"، مرتبة (تصاعديا بشكل افتراضي) حسب عمودي "البلد" و"اسم العميل":

```
SELECT * FROM Customers
ORDER BY Country, CustomerName;
```

Result

CustomerID	CustomerName	Country	City
1	Ahmed Mohammed	Iraq	Basra
3	Amer Sultan	Iraq	Baghdad
5	Zainab Mohammed	Iraq	Basra
4	Abdulrahman Saleh	Saudi Arabia	Makkah
6	Noof Saud	Saudi Arabia	Jeddah
2	Hana Saad	Syria	Damascus

D) The following SQL statement selects all customers from the "Customers" table, sorted ascending by the "Country" and descending by the "CustomerName" column:

ب - يحدد امر SQL التالي جميع العملاء من جدول "العملاء"، مرتبة تصاعديًا حسب "البلد" وتنازليًا حسب عمود "اسم العميل":

```
SELECT * FROM Customers
ORDER BY Country ASC, CustomerName DESC;
```

Result

CustomerID	CustomerName	Country	City
5	Zainab Mohammed	Iraq	Basra
3	Amer Sultan	Iraq	Baghdad
1	Ahmed Mohammed	Iraq	Basra
6	Noof Saud	Saudi Arabia	Jeddah
4	Abdulrahman Saleh	Saudi Arabia	Makkah
2	Hana Saad	Syria	Damascus

The SQL UPDATE Statement

The UPDATE statement is used to modify the existing records in a table.

يتم استخدام امر UPDATE لتعديل السجلات الموجودة في الجدول.

Syntax

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Note: Be careful when updating records in a table! Notice the WHERE clause in the UPDATE statement. The WHERE clause specifies which record(s) that should be updated. If you omit the WHERE clause, all records in the table will be updated!

ملاحظة: كن حذرًا عند تحديث السجلات في الجدول! لاحظ جملة WHERE في عبارة UPDATE. تحدد جملة WHERE السجل (السجلات) التي يجب تحديثها. إذا قمت بحذف جملةWHERE ، فسيتم تحديث جميع السجلات في الجدول!

Examples

A) The following SQL statement updates the first customer (CustomerID = 1) with a new person name and a new city:

أ - يقوم امر SQL التالي بتحديث العميل الأول (معرف العميل = ١) باسم شخص جديد ومدينة جديدة:

```
UPDATE Customers
SET CustomerName  = 'Hussein Ali', City = 'karbala'
WHERE CustomerID = 1;
```

Result

The following message appears: تظهر الرسالة التالية

```
(1 rows affected)
Completion time: 2023-10-13T21:44:13.6738293+03:00
```

To view the table records after modification, we execute the following SQL statement:

لاستعراض سجلات الجدول بعد التعديل، ننفذ امر SQL التالي:

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	Country	City
1	Hussein Ali	Iraq	karbala
2	Hana Saad	Syria	Damascus
3	Amer Sultan	Iraq	Baghdad
4	Abdulrahman Saleh	Saudi Arabia	Makkah
5	Zainab Mohammed	Iraq	Basra
6	Noof Saud	Saudi Arabia	Jeddah

B) The following SQL statement will update the customer's name to "Abdul Aziz Fahd" for all records where the country is "Saudi Arabia":

يقوم امر SQL التالي بتحديث اسم العميل إلى "عبد العزيز فهد" لجميع السجلات التي تكون فيها الدولة "المملكة العربية السعودية":

```
UPDATE Customers
SET CustomerName  = 'Abdul Aziz Fahd'
WHERE Country = 'Saudi Arabia' ;
```

Result

The following message appears: تظهر الرسالة التالية

```
(2 rows affected)
Completion time: 2023-10-13T22:33:27.6202344+03:00
```

To view the table records after modification, we execute the following SQL statement:

لاستعراض سجلات الجدول بعد التعديل، ننفذ امر SQL التالي:

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	Country	City
1	Hussein Ali	Iraq	karbala
2	Hana Saad	Syria	Damascus
3	Amer Sultan	Iraq	Baghdad
4	Abdul Aziz Fahd	Saudi Arabia	Makkah
5	Zainab Mohammed	Iraq	Basra
6	Abdul Aziz Fahd	Saudi Arabia	Jeddah

c) If you omit the WHERE clause, ALL records will be updated, So the following SQL statement will update the customer's name to "Abdul Aziz Fahd" for all records:

```
UPDATE Customers
SET CustomerName  = 'Abdul Aziz Fahd'
```

Result

The following message appears: تظهر الرسالة التالية

```
(6 rows affected)
Completion time: 2023-10-13T23:10:12.9263944+03:00
```

To view the table records after modification, we execute the following SQL statement:

لاستعراض سجلات الجدول بعد التعديل، ننفذ امر SQL التالي:

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	Country	City
1	Abdul Aziz Fahd	Iraq	karbala
2	Abdul Aziz Fahd	Syria	Damascus
3	Abdul Aziz Fahd	Iraq	Baghdad
4	Abdul Aziz Fahd	Saudi Arabia	Makkah
5	Abdul Aziz Fahd	Iraq	Basra
6	Abdul Aziz Fahd	Saudi Arabia	Jeddah

The SQL DELETE Statement

The DELETE statement is used to delete existing records in a table.

يتم استخدام امر DELETE لحذف السجلات الموجودة في الجدول.

Syntax

```
DELETE FROM table_name;
WHERE condition;
```

Note: Be careful when deleting records in a table! Notice the WHERE clause in the DELETE statement. The WHERE clause specifies which record(s) should be deleted. If you omit the WHERE clause, all records in the table will be deleted!

ملاحظة: كن حذرا عند حذف السجلات في الجدول! لاحظ جملة WHERE في عبارة DELETE. تحدد جملة WHERE السجل (السجلات) التي يجب حذفها. إذا قمت بحذف جملة WHERE، فسيتم حذف جميع السجلات الموجودة في الجدول!

Example

The following SQL statement deletes the customer "Zainab Mohammed" from the "Customers" table:

يقوم امر SQL التالي بحذف العميل "زينب محمد" من جدول "العملاء" :

```
DELETE FROM Customers
WHERE CustomerName= 'Zainab Mohammed';
```

Result

The following message appears: تظهر الرسالة التالية

```
(1 rows affected)
Completion time: 2023-10-14T00:25:44.7946797+03:00
```

To view the table records after deletion, we execute the following SQL statement:

لاستعراض سجلات الجدول بعد الحذف، ننفذ امر SQL التالي:

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	Country	City
1	Ahmed Mohammed	Iraq	Basra
2	Hana Saad	Syria	Damascus
3	Amer Sultan	Iraq	Baghdad
4	Abdulrahman Saleh	Saudi Arabia	Makkah
6	Noof Saud	Saudi Arabia	Jeddah

DELETE All Records

Deleting all rows in a table is possible without deleting the table. This means that the table structure, attributes, and indexes will be intact:

من الممكن حذف كافة الصفوف في الجدول دون حذف الجدول. وهذا يعني أن بنية الجدول وسماته وفهارسه ستكون سليمة:

Syntax 1

```
DELETE FROM table_name;
```

Or Syntax 2

```
DELETE * FROM table_name;
```

Example

```
DELETE FROM Customers;
```

```
DELETE * FROM Customers;
```

Result

The following message appears: تظهر الرسالة التالية

```
(6 rows affected)
Completion time: 2023-10-14T00:25:44.7946797+03:00
```

To view the table after deletion, we execute the following SQL statement:

لاستعراض الجدول بعد الحذف، ننفذ امر SQL التالي:

```
SELECT * FROM Customers;
```

CustomerID	CustomerName	Country	City

SQL Database Basics

المرحلة الثانية/ الفصل الدراسي الاول

قسم تقنيات أنظمة الحاسوب

المعهد التقني في البصرة

اعداد

المدرس المساعد زينب محمد جوار

السنة الدراسية: ٢٠٢٤–٢٠٢٥

The SQL SELECT TOP Clause

The SELECT TOP clause is used to specify the number of records to return.

يتم استخدام جملة SELECT TOP لتحديد عدد السجلات المراد إرجاعها.

The SELECT TOP clause is useful on large tables with thousands of records. Returning a large number of records can impact on performance.

تعتبر جملة SELECT TOP مفيدة في الجداول الكبيرة التي تحتوي على آلاف السجلات. يمكن أن يؤثر إرجاع عدد كبير من السجلات على الأداء.

Syntax

```
SELECT TOP number |  PERCENT column_name(s)
FROM table_name
WHERE condition;
```

The SQL SELECT TOP Clause Example

If we have the **Customers** table in the database **testDB:**

إذا كان لدينا جدول **العملاء** في قاعدة البيانات **testDB**:

CustomerID	CustomerName	Country	City
3	Ahmed Mohammed	Iraq	Basra
4	Hana Saad	Syria	Damascus
2	Amer Sultan	Iraq	Baghdad
1	Abdulrahman Saleh	Saudi Arabia	Makkah
6	Zainab Mohammed	Iraq	Basra
5	Noof Saud	Saudi Arabia	Jeddah

The following SQL statement selects the first four records from the "Customers" table:

يحدد امر SQL التالي السجلات الاربع الأولى من جدول "العملاء":

```
SELECT TOP 4 * FROM Customers;
```

Result

CustomerID	CustomerName	Country	City
3	Ahmed Mohammed	Iraq	Basra
4	Hana Saad	Syria	Damascus
2	Amer Sultan	Iraq	Baghdad
1	Abdulrahman Saleh	Saudi Arabia	Makkah

The SQL SELECT TOP PERCENT Example

The following SQL statement selects the first 50% of the records from the "Customers" table:

يحدد امر SQL التالي أول ٥٠٪ من السجلات من جدول "العملاء":

```
SELECT TOP 50 PERCENT * FROM Customers;
```

Result

CustomerID	CustomerName	Country	City
3	Ahmed Mohammed	Iraq	Basra
4	Hana Saad	Syria	Damascus
2	Amer Sultan	Iraq	Baghdad

The SQL SELECT TOP (Add WHERE Clause) Example

The following SQL statement selects the first two records from the "Customers" table, where the country is "Iraq":

يحدد امر SQL التالي أول سجلين من جدول "العملاء"، حيث تكون الدولة "العراق":

```
SELECT TOP 2 * FROM Customers
WHERE Country ='Iraq';
```

Result

CustomerID	CustomerName	Country	City
3	Ahmed Mohammed	Iraq	Basra
2	Amer Sultan	Iraq	Baghdad

The SQL MIN() and MAX() Functions

The MIN() function returns the smallest value of the selected column.

تقوم الدالة MIN() بإرجاع أصغر قيمة للعمود المحدد.

The MAX() function returns the largest value of the selected column.

تقوم الدالة MAX() بإرجاع أكبر قيمة للعمود المحدد.

MIN() Syntax

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

MAX() Syntax

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```

Examples

If we have the **Products** table in the database **testDB:**

إذا كان لدينا جدول **المنتجات** في قاعدة البيانات **testDB**:

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	tea	1	1	10 boxes	18
2	coffee	1	1	24 boxes	19
3	sugar	1	2	12 boxes	10
4	milk	2	2	48 bottles	21.35
5	oil	2	2	36 bottles	25

MIN() Example

The following SQL statement finds the price of the cheapest product:

يجد امر SQL التالي سعر المنتج الأرخص:

```
SELECT MIN(Price)
FROM Products;
```

Result

(No column name)
10

MAX() Example

The following SQL statement finds the price of the most expensive product:

يجد امر SQL التالي سعر المنتج الأغلى:

```
SELECT MAX(Price)
FROM Products;
```

Result

(No column name)
25

Note: You can create a name for the result column using the AS statement.

ملاحظة: يمكنك إنشاء أسم لعمود النتيجة باستخدام جملة AS .

MIN() Example

```
SELECT MIN(Price) AS SmallestPrice
FROM Products;
```

Result

SmallestPrice
10

MAX() Example

```
SELECT MAX(Price) AS LargestPrice
FROM Products;
```

Result

LargestPrice
25

The SQL COUNT(), AVG(), and SUM() Functions

The COUNT() function returns the number of rows that match a specified criteria.

ترجع الدالة COUNT() عدد الصفوف التي تطابق معايير محددة.

The AVG() function returns the average value of a numeric column.

ترجع الدالة AVG() القيمة المتوسطة لعمود رقمي.

The SUM() function returns the total sum of a numeric column.

ترجع الدالة SUM() المجموع الإجمالي لعمود رقمي.

COUNT() Syntax

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

AVG() Syntax

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

SUM() Syntax

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

Examples

COUNT() Example

The following SQL statement finds the number of products:

يجد امر SQL التالي عدد المنتجات:

```
SELECT COUNT(ProductID)
FROM Products;
```

Result

(No column name)
5

AVG() Example

The following SQL statement finds the average price of all products:

يجد امر SQL التالي متوسط سعر جميع المنتجات:

```
SELECT AVG(ProductID)
FROM Products;
```

Result

(No column name)
3

SUM() Example

The following SQL statement finds the sum of the " Price" fields in the "Product" table:

يجد امر SQL التالي مجموع حقول "السعر" في جدول "المنتجات":

```
SELECT SUM(Price)
FROM Products;
```

Result

(No column name)
93.3500003814697

SQL Database Basics

المرحلة الثانية/ الفصل الدراسي الاول

قسم تقنيات أنظمة الحاسوب

المعهد التقني في البصرة

اعداد

المدرس المساعد زينب محمد جوار

السنة الدراسية: ٢٠٢٤–٢٠٢٥

SQL INDEX

- The Index in SQL is a special table used to speed up the data search in the database tables. It also retrieves a vast amount of data from the tables frequently. The INDEX requires its own space in the hard disk.

- الفهرس في SQL هو جدول خاص يستخدم لتسريع البحث عن البيانات في جداول قاعدة البيانات. كما أنه يسترد كمية هائلة من البيانات من الجداول بشكل متكرر. يتطلب INDEX مساحة خاصة به على القرص الصلب.

- The index concept in SQL is the same as the index concept in a novel or a book.

- مفهوم الفهرس في SQL هو نفس مفهوم الفهرس في رواية أو كتاب.

- It is the best SQL technique for improving the performance of queries. The drawback of using indexes is that they slow down the execution time of UPDATE and INSERT statements. But they have one advantage also as they speed up the execution time of SELECT and WHERE statements.

- إنها أفضل تقنية SQL لتحسين أداء الاستعلامات. عيب استخدام الفهارس هو أنها تبطئ وقت تنفيذ عبارات UPDATE و .INSERTولكن لديهم ميزة واحدة أيضًا وهي تسريع وقت تنفيذ عبارات SELECTوWHERE.

- In SQL, an Index is created on the fields of the tables. We can easily build one or more indexes on a table. The creation and deletion of the Index do not affect the database's data.

- فيSQL ، يتم إنشاء فهرس في حقول الجداول. يمكننا بسهولة إنشاء فهرس واحد أو أكثر على الجدول. لا يؤثر إنشاء الفهرس وحذفه على بيانات قاعدة البيانات.

Why SQL Index?

The following reasons tell why Index is necessary in SQL:

توضح الأسباب التالية سبب ضرورة الفهرس في:SQL

1. SQL Indexes can search the information of the large database quickly.

١ - يمكن لفهارس SQL البحث في معلومات قاعدة البيانات الكبيرة بسرعة.

2. This concept is a quick process for those columns, including different values.

٢ - يعد هذا المفهوم عملية سريعة لتلك الأعمدة، بما في ذلك القيم المختلفة.

3. This data structure sorts the data values of columns (fields) either in ascending or descending order. Then, it assigns the entry for each value.

٣ - تقوم بنية البيانات هذه بفرز قيم بيانات الأعمدة (الحقول) إما بترتيب تصاعدي أو تنازلي. ثم يقوم بتعيين الإدخال لكل قيمة.

4. Each Index table contains only two columns. The first column is row_id, and the other is indexed-column.

٤ - يحتوي كل جدول فهرس على عمودين فقط. العمود الأول هو row_id، والآخر هو عمود مفهرس.

5. When indexes are used with smaller tables, the performance of the index may not be recognized.

٥ - عند استخدام الفهارس مع جداول أصغر، قد لا يتم التعرف على أداء الفهرس.

The SQL CREATE INDEX statement

The CREATE INDEX statement is used to create indexes in tables.

يتم استخدام امر CREATE INDEX لإنشاء فهارس في الجداول.

Indexes are used to retrieve data from the database very fast. The users cannot see the indexes, they are just used to speed up searches/queries.

يتم استخدام الفهارس لاسترداد البيانات من قاعدة البيانات بسرعة كبيرة. لا يمكن للمستخدمين رؤية الفهارس، فهي تستخدم فقط لتسريع عمليات البحث/الاستعلامات.

Note: Updating a table with indexes takes longer than updating a table without (because the indexes also need an update). So, only create indexes on columns that will be frequently searched against.

ملاحظة: يستغرق تحديث جدول باستخدام الفهارس وقتًا أطول من تحديث جدول بدون فهارس (لأن الفهارس تحتاج أيضًا إلى تحديث). لذلك، قم فقط بإنشاء فهارس على الأعمدة التي سيتم البحث عنها بشكل متكرر.

When should INDEXES not be used in SQL?

متى يجب عدم استخدام الفهارس فيSQL ؟

The Indexes should not be used in SQL in the following cases or situations:

لا ينبغي استخدام الفهارس في SQL في الحالات أو المواقف التالية:

- SQL Indexes can be avoided when the size of the table is small.
- When the table needs to be updated frequently.
- Indexed should not be used in those cases when the column of a table contains a large number of NULL values.

- يمكن تجنب فهارس SQL عندما يكون حجم الجدول صغيرًا.
- عندما يحتاج الجدول إلى التحديث بشكل متكرر.
- لا ينبغي استخدام المفهرس في الحالات التي يحتوي فيها عمود الجدول على عدد كبير من القيم الخالية.

CREATE INDEX Syntax

Creates an index on a table. Duplicate values are allowed:

إنشاء فهرس على الجدول. يُسمح بالقيم المكررة:

```
CREATE INDEX index_name
ON table_name (column1, column2, ...);
```

CREATE UNIQUE INDEX Syntax

Creates a unique index on a table. Duplicate values are not allowed:

إنشاء فهرس فريد على الجدول. غير خسر بالقيم المكررة:

```
CREATE UNIQUE INDEX index_name
ON table_name (column1, column2, ...);
```

Note: The syntax for creating indexes varies among different databases. Therefore: Check the syntax for creating indexes in your database.

ملاحظة: يختلف بناء جملة إنشاء الفهارس بين قواعد البيانات المختلفة. لذلك: تحقق من بناء الجملة لإنشاء الفهارس في قاعدة البيانات الخاصة بك.

CREATE INDEX Example

If we have the **Persons** table in the database **testDB:**

إذا كان لدينا جدول **الاشخاص** في قاعدة البيانات **testDB**:

ID	LastName	FirstName	Country	City
1	Mohammed	Ahmed	Iraq	Basra
2	Saad	Hana	Syria	Damascus
3	Sultan	Amer	Iraq	Baghdad

A-The SQL statement below creates an index named "idx_lastname" on the "LastName" column in the "Persons" table:

أ - يقوم امر SQL أدناه بإنشاء فهرس يسمى "idx_lastname" في عمود "LastName" في جدول "الأشخاص":

```
CREATE INDEX idx_lastname
ON Persons (LastName);
```

Result

```
Commands completed successfully.
Completion time: 2023-11-08T22:43:47.3381535+03:00
```

B-If you want to create an index on a combination of columns, you can list the column names within the parentheses, separated by commas:

ب - إذا كنت تريد إنشاء فهرس على مجموعة من الأعمدة، يمكنك إدراج أسماء الأعمدة داخل الأقواس، مفصولة بفواصل:

```
CREATE INDEX idx_pname
ON Persons (LastName, FirstName);
```

The SQL DROP INDEX statement

The DROP INDEX statement is used to delete an index in a table.

يتم استخدام امر DROP INDEX لحذف فهرس في جدول.

Syntax

```
DROP INDEX table_name.index_name;
```

SQL Database Basics

المرحلة الثانية/ الفصل الدراسي الاول

قسم تقنيات أنظمة الحاسوب

المعهد التقني في البصرة

اعداد

المدرس المساعد زينب محمد جوار

السنة الدراسية: ٢٠٢٤–٢٠٢٥

The GROUP BY statement

- In database management systems like SQL, the GROUP BY clause is a strong tool for categorizing rows of data based on one or more columns. It enables data aggregation and summarization, delivering insightful data and facilitating effective analysis. The COUNT(), MAX(), MIN(), SUM(), and AVG() aggregate functions are frequently used with the GROUP BY statement to group the result set by one or more columns. Although this feature has many advantages, it has drawbacks and particular applications.

في أنظمة إدارة قواعد البيانات مثل SQL، تُعد عبارة GROUP BY أداة قوية لتصنيف صفوف البيانات استنادًا إلى عمود واحد أو أكثر. فهي تمكن من تجميع البيانات وتلخيصها، وتقديم بيانات مفيدة وتسهيل التحليل الفعّال. تُستخدم وظائف التجميع COUNT() وMAX() وMIN() وSUM() وAVG() بشكل متكرر مع عبارة GROUP BY لتجميع مجموعة النتائج حسب عمود واحد أو أكثر. على الرغم من أن هذه الميزة لها العديد من المزايا، إلا أنها لها عيوب وتطبيقات معينة.

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

SQL GROUP BY Example 1

The following SQL statement lists the number of customers in each country:

Example مثال

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

SQL GROUP BY Example 2

The following SQL statement lists the summation of (Income) fields for each (P_ID) value:

Example مثال

```
SELECT P_ID, SUM (Income)
FROM People
GROUP BY P_ID;
```

The SQL HAVING Statement

- The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.
- Utilizing aggregate function-based conditions, HAVING clause in SQL is used to restrict the results of a query. It is frequently combined with the GROUP BY clause to filter grouped data.

– باستخدام الشروط القائمة على وظيفة التجميع، يتم استخدام شرط HAVING في SQL لتقييد نتائج الاستعلام. يتم دمجه غالبًا مع شرط GROUP BY لتصفية البيانات المجمعة.

Syntax

```
SELECT column1, column2,...
FROM table_name
GROUP BY column1, column2,...
HAVING condition
```

What is the difference between WHERE and HAVING clauses?

ما هو الفرق بين جملتي WHERE وHAVING؟

The WHERE clause is used to filter rows before they are grouped or aggregated, while the HAVING clause is used to filter grouped or aggregated results. The HAVING clause can use aggregate functions, unlike the WHERE clause.

تستخدم جملة WHERE لتصفية الصفوف قبل grouped (تجميعها) أو aggregated (تجميعها)، بينما تستخدم جملة HAVING لتصفية النتائج grouped (المجمعة) أو aggregated (المجمعة). يمكن لجملت HAVING استخدام وظائف التجميع، على عكس جملة WHERE.

SQL HAVING Example

The following SQL statement lists the number of customers in each country. Only include countries with more than 5 customers:

Example

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5;
```
