محاضرات عملي قواعد بيانات 1

المحاضرة الأولى ERD:

شرح مبسط لمخطط Entity Relationship Diagram (ERD).

تمرين:

لدى شـركة تأمين مجموعة من الزبائن، يمتلك كل منهم سـيارة أو أكثر وتهتم الشـركة بحوادث السـير التي تصيب كل سيارة من السيارات التي جرى التأمين عليها.

المطلوب إعطاء المخطط كيان – إرتباط لهذه القاعدة.

المحاضرة الثانية ERD:

تمرين:

لدى مشـفى الأسـد الجامعي عدد من المرضـى، ويعمل فيه عدد من الأطباء، يتألف المشـفى من عدد من الأقسام التخصصية، يجري قبول المريض في القسم المختص بحالته المرضية، كما يوجد بالمشفى عدد من الأقسـام المشـتركة التي تقدم الخدمات لكافة الأقسـام مثل مخبر التحاليل الطبية، قسـم التصـوير الشـعاعي، الصدلية.

يخضـــع المريض خلال إقامتة في المشـــفى لعدد من الفحوص وقد تجرى له عملية جراحية أو أكثر، لكل مريض من المرضـــى المقيمين قي قســـم معين طبيب مســـؤول عن متابعته ويكون هذا الطبيب واحد من الأطباء العاملين بالمشفى.

المطلوب:

إعطاء المخطط كيان – إرتباط لهذه القاعدة

ماذا يحصل لدى إنتقال المريض من قسم إلى آخر وكيف نمثل ذلك على قاعدة البيانات.

كيف يمكن إسترجاع السجل الطبي للمريض إذا راجع المشفى بعد مدة من خروجه من المشفى.

المحاضرة الثالثة ERD:

تمرين:

المطلوب إنشاء المخطط التصميمي لشركة توزيع أدوية تشتري هذه الشركة الأدوية من شركات تصنيع الأدوية وتخزنها في مستودعاتها الموزعة في مراكز التوزيع الأساسية في المحافظات، ومن ثم توزعها للمتاملين (صيدلية، مشفى).

لكل دواء تتعامل معه الشركة رقم مميز وإسم تجاري وإسم علمي وتاريخ إنتهاء الصلاحية.

تشتري شركة التوزيع الأدوية من الشركات المصنعة على شكل دفعات من خلال تقديم طلبات شراء للشركات المصنعة، تضم كل دفعة تتسلمها شركة التوزيع عدة أدوية موردة من شركة تصنيع محددة وتنتهي صلاحية الأدوية التي لها نفس الإسم التجاري بالدفعة الواحدة بنفس التاريخ، تسدد قيم الأدوية المستلمة بموجب فواتير (لكل فاتورة رقم ومجموعة أقلام وقيمة إجمالية).

تتلقى شركة التوزيع طلبات شراء من المتعاملين وتقبض ثمن الأدوية المباعة بموجب فواتير.

تحويل جزء من المخطط لجداول.

المحاضرة الرابعة SQL:

شرح مبسط للأدات المستخدمة Microsoft SQL Server.

شرح أنواع تعليمات :SQL

1. Data Manipulation Language -DML: SELECT, INSERT, UPDATE, DELETE.
2. Data Definition Language -DDL: CREATE, DROP, ALTER, ….
3. Data Control Language -DCL: GRANT, REVOKE.
4. Transaction Control Language -TCL: COMMIT, ROLLBACK, SAVEPOINT, …

شرح قواعد المعطيات المستخدمة: Pubs, Northwind

Pubs: وهي قاعدة معطيات لشركة تقوم ببيع الكتب. يتبع لهذه الشركة مجموعة من المتاجر Stores التي يجري كلا منها حسومات Discounts على المبيعات. يقوم كل متجر بعمليات بيع Sales للكتب Titles المتوفرة لديه. لكل كتاب مجموعة من المؤلفين TitleAuthor. كل مؤلف Author يشارك في تأليف مجموعة من الكتب. لكل كتاب دار نشر Publishers معين.في كل دار نشر

مجموعة من الوظائف Jobs التي يعمل في كل منها عدة موظفين Employees. لكل ناشر شعار وتوصيف لعنوان الناشر التي تخزن في الجدول Pub_info.

Northwind: وهي قاعدة بيانات لشركة افتراضية تدعى Northwind Traders Company. تقوم هذه الشركة بتزويد زبائنها Customers بالطلبيات Orders التي قام موظفو الشركة Employees بتوقيعها مع الزبائن. لكل طلبية مجموعة من البنود Order Items التي يوافق كل منها منتجا Product. تؤمن الشركة كل منتج عن طريق موردين Suppliers. تقوم الشركة بتوصيل الطلبية إلى الزبائن عن طريق موزعين Shippers.

المحاضرة الخامسة SQL:

لغة معالجة المعطيات **DML** Data Manipulation Language هي جزء من لغة SQL تتضمن التعليمات الخاصة باستعادة البيانات واضافتها و تعديلها وحذفها.

SELECT: وهي مخصصة لقراءة البيانات و استخلاصها من قاعدة البيانات.
INSERT: وهي مخصصة لإضافة سجلات جديدة إلى قاعدة البيانات.
DELETE: وهي مخصصة لحذف سجل أو مجموعة سجلات من قاعدة البيانات.
UPDATE: وهي مخصصة لتعديل سجل أو مجموعة من السجلات في قاعدة البيانات.

تُعتبر تعليمة SELECT من أشهر تعليمات اللغة وأكثرها استخداماً. تُستخدَم هذه التعليمة لاستعادة وانتقاء مجموعة من البيانات من قاعدة البيانات وذلك بإعادة جدول يحتوي مجموعة البيانات المطلوبة.

```
Select Col1, Col2, Col3 …
From Tab
WHERE ….
ORDER BY …
```

أمثلة:

```
Select *
From Authors
```

```
Select au_Fname
From Authors
```

```
Select au_Fname, au_lname
From Authors
```

إمكانية إعادة التسمية للحقل

```
Select au_Fname, au_lname, phone as telephone, City
From Authors
```

إمكانية دمج الحقول وإعادة التسمية

```
Select au_Fname + ' ' + au_lname as FullName, phone as telephone, City
From Authors
```

لمنع تكرار القيم

```
Select au_lname
From Authors
```

```
Select DISTINCT au_lname
From Authors
```

ترتيب النتائج

```
Select *
From titles
Order by price
```

```
Select *
From titles
Order by price, title
```

```
Select *
From titles
Order by price, title desc
```

نستخدم الكلمة المفتاحية WHERE مع تعليمة SELECT لاستعادة مجموعة من السجلات التي تحقق شرط أو مجموعة من الشروط التي نعبر عنها بعبارة شرطية. تُعيد العبارة الشرطية قيمة منطقية (صح أو خطأ).

يمكن للعبارة الشرطية أن تتضمن عمليات مقارنة مثل (= , < > , < , > , => , =<) ويتم ضم السجل الذي يحققها إلى جدول النتائج.

```
Select *
From Authors
Where state = 'CA'
Order by au_Fname, au_Lname
```

```
Select *
From Authors
Where state = 'CA' and
        City = 'Oakland'
Order by au_Fname, au_Lname
```

```
Select *
From Authors
Where (state = 'CA' and
        City = 'Oakland') or
        state = 'KS'
Order by au_Fname, au_Lname
```

المحاضرة السادسة SQL:

تُستخدم الكلمة المفتاحية LIKE ضمن العبارة الشرطية، كشرط لوجود مثيل. غالباً ما تُستخدَم هذه الكلمة مع إشارة (%)، التي تضاف إلى القيمة التي نبحث عن مثيلاتها، كبديل عن أي رقم من الأرقام أوالأحرف

Wildcard:

تستخدم مع LIKE كبديل عن محرف أو عدة محارف عند الاستعلام من قاعدة البيانات:

Wildcard	Description
%	A substitute for zero or more characters
_	A substitute for exactly one character
[charlist]	Any single character in charlist
[^charlist]	Any single character not in charlist

أمثلة:

```
Select *
From Authors
Where au_Fname Like 'an%'
```

```
Select *
From Authors
Where au_Fname Like '_n%'
```

```
Select *
From Authors
Where au_Fname Like '[jab]%'
```

```
Select *
From Authors
Where au_Fname Like '[^jab]%'
```

تُستخدم الكلمة المفتاحية BETWEEN ضمن العبارة الشرطية، كشرط لوجود قيمة محصورة بين قيمتين محددتين

```
Select *
From titles
Where price between 10 and 15
```

تُستخدم الكلمة المفتاحية IN ضمن العبارة الشرطية

```
Select *
From titles
Where price = 19.99 or price = 20
```

```
Select *
From titles
Where price IN (19.99, 20)
```

```
Select *
From titles
Where price NOT IN (19.99, 20)
```

تُستخدم الكلمة المفتاحية IS NULL ضمن العبارة الشرطية

```
Select *
From titles
Where price IS NULL
```

```
Select *
From titles
Where price IS NOT NULL
```

تُستخدم تعليمة UNION لدمج نتيجة استعلام أو أكثر ويجب مراعات ما يلي:

- Each SELECT statement within UNION must have the same number of columns
- The columns must also have similar data types
- The columns in each SELECT statement must also be in the same order

```
Use Northwind

Select city, country
from employees

Select city, country
from customers
```

لمنع التكرار

```
Select city, country
from employees
UNION
Select city, country
from customers
```

بدون حذف للتكرار

```
Select city, country
from employees
UNION ALL
Select city, country
from customers
```

من قاعدة المعطيات Northwind، أوجد المدينة والبلد التي فيها موظفين و زبائن معاً.

```
select city, country
from employees
Intersect
Select city, country
from customers
```

من قاعدة المعطيات Northwind، أوجد المدينة التي فيها موظفين ولايوجد فيها زبائن.

```
select city
from employees
except
Select city
from customers
```

المحاضرة السابعة SQL:

التوابع وتقسم إلى نوعين:

Single-row functions: These functions work on single rows only and **return** one result per row.

there is different type of single row functions

1. character function:
   which accepts character input and returns both character and number values.
   LOWER
   UPPER
   INITCAP
   CONCAT
   SUBSTR
   LENGTH
   LPAD
   RPAD
   TRIM
   REPLACE are character functions.

2. number function:
   accepts numeric input and return numeric values.

   ROUND
   TRUNC
   MOD are number functions.

3. date functions:
   date functions accepts date type and return date values except MONTHS_BETWEEN.

   MONTHS_BETWEEN
   ADD_MONTHS
   NEXT_DAY
   LAST_DAY

4. conversion functions:
   conversion functions are used to convert from one data type to another data type.

أمثلة:

CEILING: يعيد العدد الصحيح الأكبر مباشرة.

FLOOR: يعيد العدد الصحيح الأصغر مباشرة.

ROUND: يقوم بتقريب قيمة تعبير حسابي إلى الدقة المطلوبة.

```
Select price, CEILING(price), FLOOR(price), ROUND(price, 1), ROUND(price, -1)
From titles
```

LOWER: يعيد سلسلة محارف بعد تحويل الحروف الكبيرة إلى صغيرة.

UPPERE: يعيد سلسلة محارف بعد تحويل الحروف الصغيرة إلى كبيرة.

LEFT: يعيد الجزء اليساري من التعبير بطول محدد.

SUBSTRING: يُعيد جزء من سلسلة محارف، ابتداءً من موقع محدد في تلك السلسلة، وبطول عدد محدد من المحارف.

```
Select title, LOWER(title), UPPER(title), LEFT(title, 9), RIGHT(title, 9),
SUBSTRING(title, 1, 9)
From titles
```

GETDATE: يعيد التاريخ كاملاً مع الزمن.

```
Select GETDATE()
Select DATEADD(day, 3, GETDATE()) as 'new day',
       DATEADD(month, 1, GETDATE()) as 'new month',
       DATEADD(year, 1, GETDATE()) as 'new year'

Select DATENAME(year, GETDATE()) as year,
       DATENAME(month, GETDATE()) as month,
       DATENAME(day, GETDATE()) as day
```

Aggregate functions: perform a calculation on a set of values and return a single value.

- **AVG** – calculates the **average** of a set of values.
- **COUNT** – counts rows in a specified table or view.
- **MIN** – gets the minimum value in a set of values.
- **MAX** – gets the maximum value in a set of values.
- **SUM** – calculates the **sum** of values.

```
Select *
From titles
```

عدد السجلات بشكل عام

```
Select Count(*)
From titles
```

عدد السجلات التي تتضمن قيمة حسب حقل السعر

```
Select Count(Price)
From titles
```

عدد السجلات التي تتضمن قيمة حسب حقل السعر ومن دون تكرار

```
Select Count(Distinct Price)
From titles
```

مجموع المبيعات الجارية بشكل عام

```
Select SUM(YTD_SALES) 'YTD SALES'
From titles
```

مجموع المبيعات الجارية بدون تكرار

```
Select SUM(Distinct YTD_SALES) 'YTD SALES'
From titles

Select AVG(Price)
From titles

Select MAX(Price) 'Max Price', MIN(Price) 'Min Price'
From titles


Group by
Select SUM(Price) From titles Where pub_id = 0736
Select SUM(Price) From titles Where pub_id = 0877
Select SUM(Price) From titles Where pub_id = 1389

Select SUM(Price)
From titles
Group By pub_id

Select pub_id, SUM(Price)
From titles
Group By pub_id
```

Having

أوجد قائمة بالناشرين الذين تجاوزت مجموع مبيعاتهم الجارية مبلغ 25,000$.

```
SELECT PUB_ID, SUM(YTD_SALES) TOTAL
FROM TITLES
GROUP BY PUB_ID
HAVING SUM(YTD_SALES)>25000
ORDER BY PUB_ID
```

المحاضرة الثامنة SQL:

الاستعلامات الفرعية Sub Queries

```
SELECT columnA, columnB
FROM Table_Name
WHERE columnB = (Subquery)
```

```
SELECT columnA, (Subquery)
FROM Table_Name
```

بيانات الكتب التي سعرها أكبر من الكتاب Life Without Fear.

```
Select *
From titles
Where price > (Select Price
               From titles
               Where title = 'Life Without Fear')
```

قائمة بأسماء دور النشر الذين ينشرون كتبا في مجال إدارة الأعمال

```
Select PUB_NAME
From Publishers
Where PUB_ID  IN(Select PUB_ID
                  From TITLES
                  Where TYPE = 'BUSINESS')
```

قائمة بأسماء الكتب ودور نشرها

```
Select t.title, t.pub_id, (Select p.pub_name
                           From publishers p
                           Where p.pub_id = t.pub_id) AS pub_name
From titles t
```

أوجد قائمة باسماء جميع الكتب التي يساوي سعرها سعر أرخص كتاب من كتب إدارة الأعمال

```
Select type, title
From  TITLES
Where price = (Select MIN(price)
                    From TITLES
                    Where
                    type = 'business'  )
And type <> 'business'
```

التعبير EXISTS:

يُستخدم التعبير EXISTS للتحقق من إعادة الاستعلام الفرعي الذي يليه لأي سجل. ويأخذ التعبير كاملاً القيمة TRUE في حال أرجع الاستعلام الفرعي سجلاً أو أكثر، والقيمة FALSE إذا لم يُرجع الاستعلام الفرعي أي سجل.

أوجد الناشرين الذين ينشرون كتبا في مجال إدارة الأعمال.

```
Select pub_name
From publishers p
Where EXISTS (Select 1
                    From titles t
                Where t.pub_id  =  p.pub_id
                    And t.type = 'Business' )
```

أوجد المدن التي يوجد فيها مؤلفين ولا يوجد فيها ناشرين

```
Select DISTINCT city
From authors
Where NOT EXISTS
        (Select *
                From publishers
                Where  publishers.city = authors.city )
```

المحاضرة التاسعة SQL:

ربط الجداول join tables

```
Select title, pub_id
From titles

Select pub_id, pub_name
From publishers
```

الجداء الديكارتي Cartesian Product

```
Select t.title, p.pub_name
From titles t, publishers p

Select t.title, p.pub_name
From titles t Cross Join publishers p
```

الربط الداخلي Inner Join

```
Select t.title, p.pub_name
From titles t, publishers p
Where t.pub_id = p. pub_id

Select t.title, p.pub_name
From titles t
Inner Join publishers p on t.pub_id = p. pub_id

Select t.title, p.pub_name
From titles t
Inner Join publishers p on t.pub_id = p. pub_id
Where t.price > 10
And p.country = 'USA'



Select *
From titles

Select *
From titleauthor

Select *
From authors
```

```
Select t.title, a.au_fname, a.au_lname
From titles t
Inner Join titleauthor ta on ta.title_id = t.title_id
Inner Join authors  a on a.au_id = ta.au_id
```

الربط الخارجي Left, Right, Full

```
Select t.title, p.pub_name
From titles t
Left Outer Join publishers p on t.pub_id = p. pub_id

Select t.title, p.pub_name
From titles t
Right Outer Join publishers p on t.pub_id = p. pub_id

Select t.title, p.pub_name
From titles t
Full Join publishers p on t.pub_id = p. pub_id
```

المحاضرة العاشرة SQL:

DDL Commands

CREATE: وهي مخصصة لانشاء أغراض جديدة في قاعدة المعطيات.

DROP: وهي مخصصة لحذف غرض من قاعدة المعطيات.

ALTER: وهي مخصصة لتعديل غرض في قاعدة المعطيات.

```
Create Table Debt(
Deptid        int      not null,
Deptname      varchar(50)   not null
)
```

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Uniquely identifies a row/record in another table
- CHECK - Ensures that all values in a column satisfies a specific condition
- DEFAULT - Sets a default value for a column when no value is specified

```
ALTER TABLE Debt
ADD CONSTRAINT PK_Debt PRIMARY KEY (Deptid);

ALTER TABLE Debt
ADD CONSTRAINT Un_Debt Unique (Deptname);

ALTER TABLE Debt ADD col_b int NOT NULL;

ALTER TABLE Debt
ADD CONSTRAINT Ch_Debt CHECK (col_b > 0);

INSERT INTO Debt VALUES (1, 'Sales', 1)

INSERT INTO Debt VALUES (1, 'Sales', 1)
```
Violation of PRIMARY KEY constraint 'PK_Debt'. Cannot insert duplicate key in object 'dbo.Debt'. The duplicate key value is (1).

```
INSERT INTO Debt VALUES (2, 'Sales', 1)
```
Violation of UNIQUE KEY constraint 'Un_Debt'. Cannot insert duplicate key in object 'dbo.Debt'. The duplicate key value is (Sales).

```
INSERT INTO Debt VALUES (2, 'Management', 0)
```
The INSERT statement conflicted with the CHECK constraint "Ch_Debt". The conflict occurred in database "pubs", table "dbo.Debt", column 'col_b'.

```
INSERT INTO Debt VALUES (2, 'Management', 1)


Create Table Emp(
Empid       int     not null,
Empname     varchar(50)   not null,
Deptid      int
);



ALTER TABLE Emp
ADD CONSTRAINT FK_Emp FOREIGN KEY (Deptid)
   REFERENCES Debt (Deptid);

INSERT INTO Emp VALUES (1, 'Ahmad', 1)

INSERT INTO Emp VALUES (2, 'farah', 10)
```

The INSERT statement conflicted with the FOREIGN KEY constraint "FK_Emp". The conflict occurred in database "pubs", table "dbo.Debt", column 'Deptid'.

```
UPDATE Emp SET Empname = 'samer'
WHERE Empid = 1
```

```
DELETE Emp
WHERE Empid = 1
```