

قواعد بيانات

SQL

تم اعداد هذا الكتاب بواسطة:

WEBSTAR

2007

£_mail: Web_star10@yahoo.com



المقدمة:

الحمد لله رب العالمين الذي ليس لنا صفه بحمده إلا لنقوز في طاعته ولا غنى إلا في الافتقار إلى رحمته والصلاة والسلام على المبعوث بالدين والمنهج المستقيم.. على من أنار عقول البشر وأخرجهم من الظلمات إلى النور سيدنا صلى الله عليه وسلم وعلى اله وأصحابه أجمعين ومن تبعهم بإحسان إلى يوم الدين.

أما بعد:

مع كل اشراقة منيرة.. وإطلالة فجر جديد.. مع أجيال الكمبيوتر.. مع تطورات التكنولوجيا.. مع مواكبة العصر الحديث..

بحوث في آفاق الماضي وخوض في مغامرات الزمن وتحدي للسير في المستقبل الواعد..

فعلى مشارف نهاية القرن العشرين، تغيرت مفاهيم كثيرة وتطورت التكنولوجيا بصورة مذهلة بحيث لم يعد الإنسان قادراً على مواكبة تطورات العصر بالطريقة العادية التي كان يتبعها وكان لابد له من الاستعانة بشيء يسهل له هذه المهمة حتى لا يفوته قطار المعرفة والتقدم والنجاح.

وانه من جملة التطورات التي حدثت في العشر الأخيرة من هذا القرن ولحسن الحظ وهو تطور الكمبيوتر الذي أصبح بمتناول الجميع بعد ما كان استخدامه محصوراً ومقتصرًا على الشركات والمؤسسات العملاقة وأصبح الإنسان في يومنا الحاضر بحاجة ماسة إلى فهم مبادئ هذا الاختراع على الأقل والقدرة على استخدامه في زمن أصبح فيه الجاهل بمبادئ الكمبيوتر مثل الأمي الجاهل بحروف الهجائية.

تلخيص هذا الكتاب تم من ترجمة بعض الكتب الالكترونية الانجليزية وايضا مقتبسه من مواقع تعليمية وكتب الكترونية عربية عديدة وايضاً هو مخلص بالدروس المأخوذة

من الدكتور القدير/ طاهر الرشاحي

وانا شخصياً اعتبر هذا الكتاب هو مرجعي الوحيد في أيام الاختبارات النهائية وان شاء الله بالنجاح للجميع...

كلمة المؤلف:

هذا الكتاب تجربة بسيطة لي يحتوي على بعض الأخطاء الغير مقصودة فالإنسان دائما ما
يخطئ "جل من لا يخطئ"
وهذا الكتاب مازال قيد التطوير
فمن وجد بعض الأخطاء عليه تبليغي عبر بريدي الالكتروني ومن لديه أي ملاحظات أو
إضافات لهذا الكتاب فلا يتردد بعمل الخير لإخوانه

مع تحياتي

WebStar

e-mail: web_star10@yahoo.com

Mobile: 711273934



خاص بـسـنـجـم الـوـيـب

كلية علوم وهندسة الحاسوب

مستوى ثاني

جامعة القادسية



*اهدي هذا الكتاب إلى..
الذان وصانا بهما رب العالمين ورسولنا الأمين
محمداً صلى الله عليه وسلم
إلى أبي الغالية التي نورت لي الطريق أمامي،،
إلى والدي العزيز الذي له المقام الأول
لتشجيعي لعمل الخير والمثابرة بالاجتهاد ،،،
*اهدي هذا الكتاب إلى أصدقائي الذين قاموا بتشجيعي
ودعمني بعمل وإنزال هذا الكتاب بثقة واقتدار
*اهدي هذا الكتاب إلى زملائي وزميلاتي
في كلية علوم وهندسة الحاسوب كافة،،
*إهدائي الخاص بالدركتور التقرير : طاهر الرشاحي
الذي حفزنا لحب هذه المادة واللغة خاصة.

تحياتي،،،

كيفية التعامل مع لغة التوصيف ويمكنك ذلك عن طريق تجربة تنفيذ بعض الأوامر والأمثلة التي هنا على برنامج الاستعلامات المرفق مع حزمة أوراكل

إنشاء جدول

إنشاء جدول قاعدة بيانات لحفظ البيانات والتعامل معها

| | |
|--------------|------------|
| CREATE TABLE | إنشاء جدول |
|--------------|------------|

التوصيف

CREATE TABLE (طول الحقل نوع الحقل اسم الحقل) اسم الجدول

مثال

Sq|>CREATE TABLE PHONE (PERSON_NAME CHAR (30),

2 PERSON_NO NUMBER (6) PRIMARY KEY, PERSON_ADD VARCHAR2(100),

3 TEL_NO NUMBER (7) NOT NULL)

أنشأنا في المثال السابق جدول وإنشاءنا به Phone
الحقول التالية أسمه

| التعريف | النوع | الطول | الأسم | ملاحظات |
|-------------|-------|-------|-------------|----------------------|
| اسم الشخص | حرفي | 30 | PERSON_NAME | - |
| رقم الشخص | رقمي | 6 | PERSON_NO | حقل مفتاح أساسي |
| عنوان الشخص | حرفي | 100 | PERSON_ADD | - |
| الهاتف رقم | رقمي | 7 | TEL_NO | لا يمكن أن يكون فارغ |

ملاحظات

(PRIMARY KEY)

عند وضع كلمة فهذا يعني أن الحقل حقل مفتاح أي انه يتميز عن الحقول الأخرى ولا يمكن أن يكون في كل ا جدول سوى حقل مفتاح واحد

| خصائص حقل المفتاح | | |
|----------------------------|----------|---------|
| وحيد لا يسمح بتكرار القيمة | رقمي | مفهرس |
| Unique | Not Null | Indexed |

(NOT NULL)

عند وضع هذه الكلمة هذا يعني أن الحقل لا يمكن أن يكون فارغ أبدا

لحذف جدول منشاء

اسم الجدول Drop table

مثال:

```
SQL> drop table phone;
```

Table dropped.

تنبيه

في حال تم حذف الجدول فسوف يتم إزالته نهائيا مع كافة البيانات التي به ولا يمكن التراجع عن هذه العملية

إضافة حقل في جدول منشاء

إضافة حقول إلى جدول

Alter table اسم الجدول Add (الحقول المضافة)

مثال:

```
SQL> alter table phone add (fax_no number(9));
```

Table altered.

وسيصبح الان الجدول الذي لدينا هو:

```
SQL> desc phone;
Name                               Null?   Type
-----
PERSON_NAME                       CHAR(30
PERSON_NO                         NOT NULL NUMBER(6
PERSON_ADD                        VARCHAR2(100
TEL_NO                           NOT NULL NUMBER(7
FAX_NO                           NUMBER(9
```

تعديل بيانات حقول

لإجراء تعديل على نوع الحقل المعين مباشرة..

Alter table اسم الجدول modify (اسم الحقل مع التعديلات الجديدة)

مثال:

```
SQL> alter table phone modify (tel_no number(9));
```

Table altered.

تنبيه

لا يمكن تعديل اسم الحقل ولكن يمكن تعديل نوعه وطوله ومواصفاته

عمليات على الجداول

لإنشاء جدول جديد من بنية جدول آخر

```
SQL> create table phone2 as select *from phone;
```

توضيح

تم عمل نسخة من الجدول الأصلي بنسخة جديدة مع تغيير الأسم.

لربط جدول أو عدة جداول مع بعضها البعض

عند عملية الإنشاء أو في عملية التعديل الخاصة بالجداول يتم عمل الأتي

CREATE TABLE اسم الجدول REFERENCES (اسم الحقل) اسم الجدول

مثال

```
Sql>CREATE TABLE PHONE (PERSONE_NO number(6) REFERENCES PERSONE
```

```
2 (PERSONE_NO);
```

توضيح

تم ربط الجدول الأول مع الجدول الثاني عن طريق حقل أسم الشخص

تنبيه

يجب أن تكون بيانات الحقول المرتبطة في ما بينها متشابهة من حيث النوع والطول.

المحددات:

هناك مجموعة من المحددات يمكننا استخدامها على الحقول وهي كالتالي:

- **Not Null** يجب إدخال قيمة في الحقل
- **Unique** أن لا تكرر قيمة الحقل
- **Default** وضع قيمة افتراضية للحقل
- **Check** إجراء فحص معين على الحقل
- **Primary Key** إنشاء قيد مفتاح رئيسي
- **Foreign Key** إنشاء قيد مفتاح ثانوي

لإنشاء المحددات بكل سهولة ولكن تكون بطريقتين:

- ١- إنشاء المحدد على مستوى **تعريف الحقل** ، أي يُكتب المحدد في نفس سطر تعريف الحقل ، أو في السطر التالي مباشرة..
 - ٢- إنشاء محدّدات على مستوى **تعريف الجدول** ، أي تكتب جميع المحددات بعد الإنتهاء من تعريف الحقول ، وهذه الطريقة هي المعتمدة من الشركة ، وهي الأفضل حسب رأي الكثيرين..
- وننبه** هنا إلى أن هاتين الطريقتين تسريان على جميع المحددات الستة التي ذكرناها ، ماعدا المحدد ذو القيمة الافتراضية للحقل (**Default**) فتسري عليه الطريقة الأولى فقط.

والآن ماعلينا الا شرح هذه المحددات:

١- المحدد الاول (Not Null)

وهو وضع قيد على حقل ما ، بحيث لا يكون هذا الحقل ذو قيمة فارغة.. أي يجب أن يحتوي على قيمة .. كحقل رقم الطالب وحقل اسم الطالب مثلاً ، يجب أن يحويان على قيمة..

مثال بالطريقة الاولى:

```
SQL> create table customer (  
2 cust_no number(4) not null,  
3 cust_name varchar2(40) not null,  
4 cust_address varchar2 (30)  
5 );
```

التوضيح:

* أنشأنا هنا الجدول الخاص بالزبائن ؛ أن الحقل الأول : رقم الزبون (cust_no) والحقل الثاني : اسم الزبون (cust_name) كلاهما عرفناهما بأن لا يكونا ذو قيمة فارغة.. لأنه يجب معرفة رقم الزبون واسمه .. أما الحقل الثالث : عنوان الزبون ، (cust_address) فليس مهماً ، فلا ضير أن يكون فارغاً عند إدخال البيانات.

* أننا ذكرنا اسم القيد وهو not null بعد تعريف الحقل مباشرة ، سواء في نفس السطر كما في تعريف حقل رقم الزبون ، أو في سطر آخر كما في تعريف حقل اسم الزبون.

بالطريقة الثانية:

```
SQL> create table customer (  
2 cust_no number(4),  
3 cust_name varchar2(40) ,  
4 cust_address varchar2 (30),  
5 constraint cust_no_nt  
6 check (cust_no is not null),  
7 constraint cust_name_nt  
8 check ( cust_name is not null)  
9 );
```

أننا عرفنا الحقول أولاً ، ثم وضعنا القيود ، وهذه الطريقة أفضل من الأولى ؛ فمثلاً عندما عرفنا الحقل (رقم الزبون) كتبنا الآتي..
حيث ان constraint كلمة محجوزة معناها قيد و cust_no_nt اسم القيد
أما (cust_no is not null) check فمعناه افحص الحقل cust_no هل ليس فارغاً.. وهذا ما نريده أن لا يكون فارغاً) أي يحتوي على قيمة. (

٢-المحدد الثاني (Unique):

ومعناه ألا تتكرر قيمة هذا الحقل الذي سنضع هذا القيد عليه ، فمثلاً لو وضعنا هذا القيد على حقل رقم الزبون ، نستنتج أنه لا نريد أن يتكرر رقم الزبون.. كما يلي:

مثال للطريقة الاولى:

```
SQL> create table customer (  
2 cust_no number(4) unique,  
3 cust_name varchar2(40) not null,  
4 cust_address varchar2(30)  
5 );
```

مثال للطريقة الثانية:

```
SQL> create table customer (  
2 cust_no number(4),  
3 cust_name varchar2(40) not null,  
4 cust_address varchar2(30),  
5 constraint cust_no_uni  
6 unique ( cust_no)  
7 );
```

٢-١ المحدد الثالث: (Default):

ونستفيد منه في وضع قيمة افتراضية لحقل ما ، مثلاً لحقل عمر الطالب ، نضع قيمة افتراضية لعمره ، وذلك في حالة عدم إدخال المستخدم أي قيمة.. فقط وهذا القيد هو الوحيد الذي يكتب بطريقة .الاولى فقط:
مثال:

```
create table student (  
std_no number(7) not null,  
std_name varchar2(40) not null,  
std_age number(2)  
default 20 ,  
std_nation varchar2(20)  
default 'Yamani'  
);
```

نلاحظ:

عرفنا حقل عمر الطالب (std_no) على أنه رقم وأعطيناه قيمة افتراضية في حالة أننا عدم إدخال المستخدم لأي قيمة ، وهي القيمة □سنة□20
و عرفنا حقل جنسية الطالب (std_nation) على أنه نصي ، وأعطيناه قيمة افتراضية في حالة عدم إدخال المستخدم لأي قيمة ، وهي الجنسية 'Yemeni'

٢-٢ المحدد الثالث (CHECK):

ونستفيد منه عندما نريد أن نفحص قيمة مدخلة لحقل معين يقبل مجموعة قيم محددة سلفاً، حيث يقوم القيد بفحص القيمة المخلة من بين القيم الموجودة.

مثال بالطريقة الاولى:

```
SQL> create table student (  
2 std_no number(7) not null,  
3 std_name varchar2(40) not null,  
4 std_sex varchar2(1)  
5 check ( std_sex in (' m ',' f ')),  
6 std_case varchar2(1)  
7 check ( std_case in (' s ',' m ',' w ',' d ')),  
8 std_age number(2)  
9 check ( std_age between 19 and 30 )  
10 );
```

أننا عرفنا حقل جنس الطالب (std_sex) على أنه نصي وأعطيناه قيد الفحص check□□□ هنا يفحص قيمة الحقل ، فعندما يدخل المستخدم الحرف m .. يعني ذلك أن الطالب ذكر عندما يدخل المستخدم الحرف f يعني ذلك أن الطالب أنثى

وعرفنا حقل الحالة الإجتماعية للطالب (std_case) على أنه نصي وأعطيناه قيد هنا يفحص قيمة الحقل ، فعندما يدخل المستخدم الحرف فهذا يفحص check الفحص قيمة الحقل..
 فعندما يدخل المستخدم الحرف s يعني ذلك أن الطالب متزوج
 وعندما يدخل المستخدم الحرف w يعني ذلك أن الطالب أرمل.
 وعندما يدخل المستخدم الحرف m يعني ذلك أن الطالب مطلق
 وعندما يدخل المستخدم الحرف d يعني ذلك أن الطالب أعزب.
 وعرفنا حقل عمر الطالب (std_age) على أنه نصي وأعطيناه قيد الفحص check الفحص قيمة الحقل..
 يفحص قيمة الحقل ما بين القيمة العمرية من 19 إلى 30

مثال للطريقة الثانية:

```
SQL> create table student
2 ( std_no number(7) not null,
3 std_name varchar2(40) not null,
4 std_sex varchar2(1),
5 std_case varchar2(1),
6 std_age number(2),
7 constraint std_sex_chk
8 check (std_sex in (' m ',' f ')),
9 constraint std_case_chk
10 check( std_case in (' s ',' m ',' w ',' d ')),
11 constraint stdstd_age_chk
12 check ( std_age between 19 and 30)
13 );
```

هـ- المحدد الخامس (Primary Key):

وظيفة هذا القيد إعطاء حقل معين من عدة حقول في جدول ما ؛ صفة المفتاح الرئيسي في هذا الجدول..
 المفتاح الذي يحدد بشكل وحيد ومتفرد بحيث يتميز عن غيره ، فلا تتكرر قيمته في أكثر من حقل واحد ،
 ولا يقبل قيم (Null) أي لا يمكننا أن نترك الحقل فارغاً بدون قيمة

مثال للطريقة الاولى:

```
SQL> create table student
2 ( std_no number(10)
3 primary key,
4 std_name varchar2(50)
5 );
```

الطريقة الثانية:

```
SQL> create table student
2 ( std_no number(10) ,
3 std_name varchar2(50),
4 constraint pk_std_01
5 primary key (std_no)
6 );
```

٦-المحدد السادس (Forgein Key):

وظيفة هذا القيد إعطاء حقل معين من عدة حقول في جدول ما ؛ صفة المفتاح الأجنبي في هذا الجدول. عبارة عن حقل) صفة (أو أكثر يستخدم للربط بين جدولين ، وسمي المفتاح الأجنبي بهذا الاسم لأنه ليس من الحقول الموجودة أصلاً في الجدول ، أي أنه عبارة عن حقل أو أكثر تضاف إلى جدول لربطه مع جدول آخر.

ويستخدم المفتاح الأجنبي كمؤشر مقابل للمفتاح الرئيسي ، بمعنى آخر فإن المفتاح الأجنبي هو عبارة عن حقل) صفة (أو أكثر تضاف لج دول لربطه مع جدول آخر ، مع الالتزام بوجود مفتاح رئيسي مقابل مع ملاحظة أن المفتاح الأجنبي يجب أن يكون من نفس نوع بيانات (Type Data) المفتاح الرئيسي ، فلو كان المفتاح الرئيسي من النوع رقم مثلاً ، يجب أن يكون المفتاح الأجنبي من النوع رقم ، لذلك يعتبر الأجنبي مؤشر للرئيسي.

مثال بالطريقة الاولى:

حتى ننشئ مفتاحاً أجنبياً في جدول ما ، يجب أن يكون هذا المفتاح رئيسياً في جدول آخر أنشأناه سابقاً ، وفي هذا المثال سنقوم إن شاء الله بإنشاء مفتاح رئيسي في جدول الأقسام وهو رقم القسم ، ثم نقوم بإنشاء حقل أجنبي في جدول الطلاب يعود لهذا الرئيسي ، وبالمثال يتضح المقال!

إنشاء جدول الأقسام ، ويسمى هذا الجدول بالرئيسي (Mister) أو الأب..

```
SQL> create table section
2 ( sec_no number(2)
3 primary key ,
4 sec_name varchar2(20) not null
5 );
```

..إنشاء جدول الطلاب ، ويسمى هذا الجدول بالتفصيل (Detail) أو الابن

```
SQL> create table student (
2 std_no number(7)
3 primary key,
4 std_name varchar2 (30) not null,
5 sec_no number(2) references
6 section (sec_no)
7 );
```

أننا عرفنا حقل واسمه رقم القسم (sec_no) الذي باللون الأخضر (وهذا أمر ضروري جداً أن نعرف الأجنبي أولاً في الجدول التفصيل ، ثم نبين أنه مؤشر إلى مفتاح رئيسي في جدول آخر بحجم هو نفس حجم رقم القسم في جدول الأقسام .. ،لأننا نعلم أن المفتاح الأجنبي يجب أن يكون تعريفه من نفس نوع بيانات وحجم المفتاح الرئيسي الذي يشير إليه ..ولكن ليس شرطاً أن يكون بنفس الاسم.. كلمة references والتي تعني أنه يشير إلى المفتاح الرئيسي في جدول الأقسام كتبنا بطريقة Section وهو المفتاح(sec_no) باللون الرمادي (ولا ننسى أن ماسبق كله هو بالطريقة الاولى.

أما المفتاح الأجنبي فسننشئه إن شاء الله بالطريقة الثانية

```
SQL> create table student (  
2 std_no number(7)primary key,  
3 std_name varchar2 (30) not null,  
4 sec_no number(2),  
5 std_age number(2),  
6 constraint fk_std_01  
7 foreign key (sec_no)  
8 references section (sec_no)  
9 );
```

نلاحظ أنه بالإضافة إلى وجود تعريف للمفتاح الرئيسي في هذا الجدول وهو رقم الطالب ، (std_no) احتوى هذا الجدول أيضاً مفتاحاً ثانوياً وهو رقم القسم (sec_no) فيسمح تعريف المفتاح الثانوي بتحديد الحقل المرجع في الجدول المرتبط به ، والذي يمكن أن يأخذ اسماً آخر.. ولكي يكون تعريف المفتاح الثانوي سليماً ، يجب أن يكون كلا الحقليْن من نفس نوع البيانات ونفس الحجم كما ذكرنا سابقاً).. جرب تعريف الحقل الثانوي بنوع آخر أو بحجم يختلف عن الحقل المؤشر إليه. (

وعند تعريف المفتاح الثانوي فإننا نخبر أورا كل بأننا نريد إنشاء تكامل مرجعي بين الحقل رقم القسم (sec_no) في الجدول student .والجدول section . وهذا الأمر يمنع الحقل في جدول الإبن student من احتواء قيمة غير موجودة في العمود : ، المرتبط به في الجدول الأب : section □

لإنشاء اسم مستخدم جديد:

* يوجد في بيئة (SQL * Plus) مستخدمين افتراضيين هما : system بكلمة السر : manager و scott بكلمة السر : tiger ..
صيغة إنشاء مستخدم في بيئة (SQL * Plus) هي:

```
SQL> create user name user identified by password ;
```

حيثُ **name user** اسم للمستخدم و **password** . كلمة السر الخاصة به.
* لايسمح بإنشاء مستخدم ومنحه صلاحيات إلا بعد الاتصال بالنظام ، ويكونُ الاتصال بالنظام و بأي مستخدم بالصيغة التالية..

```
SQL> connect اسم المستخدم
```

* لإعطاء صلاحية الاتصال لمستخدم ، نستخدم الصيغة التالية .

```
SQL> grant connect to name user ;
```

حيثُ name user .. اسم للمستخدم
الآن ليس لكم الاصلاحية الإتصال فقط ، ولكي تكون ليكم باقي الصلاحيات ، نكتب الأمر التالي..

```
SQL> grant resource to اسم المستخدم ;
```

فظهر لنا العبارة التالية ، والتي تخبرنا أنه تم منح الصلاحية..

Grant succeeded.

والآن بإمكانكم إنشاء الجداول والتعامل معها بكل حرية من خلال اسمك الخاص..

التقارب SELECT :-

المعالجة تكون داخل SELECT كيف تربط بين اكثر من جدول..

البنية:

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM table;
```

حيث ان رمز * يشير الى كل البيانات في الجدول

DISTINCT: ازالة البيانات المكرره

column: اسم الحقل

alias: المنطقة التي ينتمي اليها.

Table: يشير الى اسم الجدول

مثال:

```
Sql>SELECT *FROM departments;
```

في مثالنا السابق يظهر لك الجدول بجميع بياناته وحقله

مثال اخر:

```
Sql>SELECT department_id, location_id
2 FROM departments;
```

هذا المثال اعطانا تقرير عن الجدول بحقلين اثنين فقط رقم القسم وموقعه .

مثال :

باستطاعتنا عمل عملية حسابية كمافي المثال التالي:

```
Sql>SELECT last_name, salary, salary + 300
2 FROM employees;
```

في مثالنا السابق يظهر لنا تقرير باسم الموظف وراتبه وراتبه+٣٠٠ أي ثلاثه حقول.

اولوية العمليات الحسابية:

*الضرب والقسمة تأخذان الاولوية على الجمع والطرح.
*إذا تلاقى عمليتين بنفس الاولوية نبدأ من اليسار الى اليمين.
*الاقواس يأخذان الاولوية قبل العمليات الحسابية.

استعمال عناوين مستعاره لعناوين الاعمدة:

١- لعمل اسم مستعار لعمود ما يكون اسم العمود يليه الكلمة AS يليها الاسم المستعار.

مثال:

```
Sql>SELECT last_name AS name, commission_pct AS comm FROM employees;
```

٢- بطريقة اخرى نعمل اسم العمود يليه ف Space راغ يليه "الاسم المستعار" بين الحاصرتين.

مثال:

```
Sql>SELECT last_name "Name",  
2 salary*12 "Annual Salary" FROM employees;
```

لدمج حقليين في حقل واحد:

```
Sql>SELECT last_name||job_id AS "Employees"  
2 FROM employees;
```

استعمال سلسلة من الحروف ل اظهارها داخل الحقول المدمجة:

```
Sql>SELECT last_name ||' is a '||job_id AS "Employee Details"  
2 FROM employees;
```

| Employee Details |
|---------------------|
| King is a AD_PRES |
| Kochhar is a AD_VP |
| De Haan is a AD_VP |
| Hunold is a IT_PROG |
| Ernst is a IT_PROG |

مثال لازالة ارقام الاقسام المكرره بالدالة DISTINCT :

```
Sql>SELECT DISTINCT department_id FROM employees;
```

تقارير مشروطة:

البنية:

```
SELECT *|{[DISTINCT] column|expression [alias],...}  
FROM table  
[WHERE condition(s)];
```

البنية مشروحة سابقاً

مثال لاطهار تقرير عن رقم الموظف واسمه وقسمه لجدول الموظفين عندما رقم القسم ٩٠

```
Sql>SELECT employee_id, last_name, job_id, department_id
2      FROM employees
3      WHERE department_id = 90;
```

مثال لاطهار اسم الموظف ووظيفته وقسمه للموظف الذي اسمه 'Goyal'.

```
Sql>SELECT last_name, job_id, department_id
2      FROM employees
3      WHERE last_name = 'Goyal';
```

مثال لاطهار اسم الموظف وراتبه للموظفين الذين راتبها اقل من او يساوي ٣٠٠٠

```
Sql>SELECT last_name, salary
2      FROM employees
3      WHERE salary <= 3000;
```

العمليات الحسابية المستخدمة:

| العملية | عملها |
|---------|------------------|
| = | يساوي |
| > | اكبر من |
| >= | اكبر من او يساوي |
| < | اقل من |
| <= | اقل من او يساوي |
| <> | لا يساوي |

شروط المقارنة الاخرى:

| رقم الشرط | الشرط | عملة |
|-----------|-------------------|-------------------------|
| 1 | BETWEEN ...AND... | ما بين قيمتين (شامله) |
| 2 | IN(set) | اخذ من قيم معينه |
| 3 | LIKE | البحث عن حروف في الجدول |
| 4 | IS NULL | عندما تكون القيمة خالية |

الامثلة التي عليها هي..

مثال على الشرط رقم ١:

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500;
```


مثال على الشرط رقم ٢:

```
Sql>SELECT employee_id, last_name, salary, manager_id
2     FROM employees
3     WHERE manager_id IN (100, 101, 201);
```

الخرج سيكون:

| EMPLOYEE_ID | LAST_NAME | SALARY | MANAGER_ID |
|-------------|-----------|--------|------------|
| 202 | Fay | 6000 | 201 |
| 200 | Whalen | 4400 | 101 |
| 205 | Higgins | 12000 | 101 |
| 101 | Kochhar | 17000 | 100 |
| 102 | De Haan | 17000 | 100 |
| 124 | Mourgos | 5800 | 100 |
| 149 | Zlotkey | 10500 | 100 |
| 201 | Hartstein | 13000 | 100 |

مثال على الشرط رقم ٣:

```
Sql>SELECT first_name
2     FROM employees
3     WHERE first_name LIKE 'S%';
```

في مثالنا السابق سيكون البحث على كلمة تنتهي بحرف S

| الرمز | كيفية البحث |
|-------|---|
| 'A%' | يكون البحث في نهاية الكلمة حرف A |
| '%A' | يكون البحث في بداية الكلمة حرف A |
| '%A%' | يكون البحث اذا كانت الكلمة تبدأ وتنتهي بحرف A |
| '_A%' | يكون البحث ثاني حرف في الكلمة A |

مثال على الشرط رقم ٤:

```
Sql>SELECT last_name, manager_id
2     FROM employees
3     WHERE manager_id IS NULL;
```

عن يكون رقم مديره قيمة خالية.

شروط منطقية اخرى:

الشرط المنطقي AND: ويتحقق عند يكون كلا الشرطين محققين

مثال:

```
Sql>SELECT employee_id, last_name, job_id, salary
2   FROM employees
3   WHERE salary >=10000
4   AND job_id LIKE '%MAN%';
```

استعلام عندما يكون الراتب اكبر من او يساوي ١٠٠٠٠ و له وظيفه فيها كلمة MAN

الشرط المنطقي OR: ويتحقق الشرط عندما يكون واحد على الاقل من الشروط محقق.

مثال:

```
Sql>SELECT employee_id, last_name, job_id, salary
2   FROM employees
3   WHERE salary >= 10000
4   OR job_id LIKE '%MAN%';
```

استعلام عندما يكون الراتب اكبر من او يساوي ١٠٠٠٠ او له وظيفة تحتوي كلمة MAN

الشرط المنطقي NOT: يعيد قيمة صحيحة عندما يكون الشرط خاطئاً:

مثال

```
Sql>SELECT last_name, job_id
2   FROM employees
3   WHERE job_id NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP');
```

استعلام عند الموظفين الذي ليس لهم هذه الوظائف: **'IT_PROG', 'ST_CLERK', 'SA_REP'**

الاولويات كاملة:

| Order Evaluated | Operator |
|-----------------|-------------------------------|
| 1 | Arithmetic operators |
| 2 | Concatenation operator |
| 3 | Comparison conditions |
| 4 | IS [NOT] NULL, LIKE, [NOT] IN |
| 5 | [NOT] BETWEEN |
| 6 | NOT logical condition |
| 7 | AND logical condition |
| 8 | OR logical condition |

العملية ORDER BY:

هو البند الذي يعمل على الترتيب:
١- ASC ترتيب تصاعدي وهو الترتيب الافتراضي .

٢- DESC ترتيب تنازلي

ملاحظة : في حالة لم نحدد له أي من الكلمتين اللغة تعتمد الاول مباشرة ASC.

البنية:

```
SELECT *|{[DISTINCT] column|expression [alias],...}  
FROM table  
[WHERE condition(s)]  
[ORDER BY {column, expr, alias} [ASC|DESC]];
```

مثال:

```
SQL>SELECT last_name, job_id, department_id, hire_date  
2      FROM employees  
3      ORDER BY hire_date;
```

في مثالنا السابق يعمل على ترتيب تصاعدي لانه لم يحدد لنا في البند أي عملية نعملها فيعتمد الاول.

مثال اخر:

```
SQL>SELECT last_name, job_id, department_id, hire_date  
2      FROM employees  
3      ORDER BY hire_date DESC;
```

في مثالنا السابق يعمل على الترتيب التنازلي لانه حدد لنا البند DESC

مثال لترتيب اكثر من عمود:

```
sql>SELECT last_name, department_id, salary  
2      FROM employees  
3      ORDER BY department_id, salary DESC;
```

في مثالنا السابق قام بترتيب عمودي رقم القسم والراتب.

الوظائف الخاصة بتحويل الحروف:

| Function | Result |
|-----------------------|------------|
| LOWER('SQL Course') | sql course |
| UPPER('SQL Course') | SQL COURSE |
| INITCAP('SQL Course') | Sql Course |

وظائف اخرى خاصة بالحروف:

| Function | Result |
|-----------------------------|------------|
| CONCAT('Hello', 'World') | HelloWorld |
| SUBSTR('HelloWorld',1,5) | Hello |
| LENGTH('HelloWorld') | 10 |
| INSTR('HelloWorld', 'W') | 6 |
| LPAD(salary,10,'*') | *****24000 |
| RPAD(salary, 10, '*') | 24000***** |
| TRIM('H' FROM 'HelloWorld') | elloWorld |

مثال على ماسبق:

```
Sql>SELECT employee_id, CONCAT(first_name, last_name) NAME, job_id,  
2     LENGTH (last_name), INSTR(last_name, 'a') "Contains 'a'?"  
3     FROM employees  
4     WHERE SUBSTR(job_id, 4) = 'REP';
```

سؤال: ماذا سيكون الخرج على هذا المثال؟

استخدام دوال التاريخ:

MONTHS_BETWEEN ('01-SEP-95','11-JAN-94')→ 19.6774194

- ADD_MONTHS ('11-JAN-94',6)→ '11-JUL-94'
- NEXT_DAY ('01-SEP-95','FRIDAY')→ '08-SEP-95'
- LAST_DAY('01-FEB-95')→ '28-FEB-95'

Assume SYSDATE = '25-JUL-95':

- ROUND(SYSDATE,'MONTH')→ 01-AUG-95
- ROUND(SYSDATE , 'YEAR') → 01-JAN-96
- TRUNC(SYSDATE , 'MONTH')→ 01-JUL-95
- TRUNC(SYSDATE , 'YEAR') → 01-JAN-95

عملية ربط الجداول:

الربط المتساوي:

هي عملية لربط جدولين بواسطة المفتاح الاساسي والمفتاح الاجنبي.

البنية:

```
Sql>SELECT table1.column, table2.column  
2     FROM table1, table2  
3     WHERE table1.column1 = table2.column2;
```

الصورة التوضيحية

| EMPLOYEE_ID | DEPARTMENT_ID |
|-------------|---------------|
| 200 | 10 |
| 201 | 20 |
| 202 | 20 |
| 124 | 50 |
| 141 | 50 |
| 142 | 50 |
| 143 | 50 |
| 144 | 50 |
| 103 | 60 |
| 104 | 60 |

| DEPARTMENT_ID | DEPARTMENT_NAME |
|---------------|-----------------|
| 10 | Administration |
| 20 | Marketing |
| 20 | Marketing |
| 50 | Shipping |
| 50 | Shipping |
| 50 | Shipping |
| 50 | Shipping |
| 50 | Shipping |
| 50 | Shipping |
| 60 | IT |
| 60 | IT |

19 rows selected

Foreign key

Primary key

مثال:

```
Sql>SELECT employees.employee_id, employees.last_name,  
2     employees.department_id, departments.department_id,  
3     departments.location_id  
4     FROM employees, departments  
5     WHERE employees.department_id = departments.department_id;
```

الربط الخارجي: تتميز عملية الربط الخارجي الاشارة +

البنية:

```
Sql>SELECT table1.column, table2.column  
2     FROM table1, table2  
3     WHERE table1.column (+) = table2.column;
```

مثال:

```
Sql>SELECT e.last_name, e.department_id, d.department_name
2      FROM employees e, departments d
3      WHERE e.department_id(+) = d.department_id;
```

الربط الخارجي من اليسار:

مثال:

```
Sql>SELECT e.last_name, e.department_id, d.department_name
2      FROM employees e
3      LEFT OUTER JOIN departments d
4      ON (e.department_id = d.department_id);
```

الربط الخارجي من اليمين:

مثال:

```
Sql>SELECT e.last_name, e.department_id, d.department_name
2      FROM employees e
3      RIGHT OUTER JOIN departments d
4      ON (e.department_id = d.department_id);
```

الربط مع نفسه:

مثال:

```
Sql>SELECT worker.last_name || ' works for ' || manager.last_name
2      FROM employees worker, employees manager
3      WHERE worker.manager_id = manager.employee_id;
```

استرجاع السجلات مع الربط الطبيعي:

```
Sql>SELECT department_id, department_name, location_id, city
2      FROM departments
3      NATURAL JOIN locations;
```

دالة التجميع Group By & Having

تستخدم دالة التجميع هذه في حالات نحتاج بها إلى الحصول على معلومات مقسمة على أساس معين فمن الممكن استخدامها في الحصول على بيانات موظفين دائرة معينة كل على حدى وتستخدم في ذلك بعض الدالات الرياضية والتي نحتاجها للحصول على نتائج معينة من كل قسم مثل المتوسط الحسابي لأعمار الموظفين في كل دائرة

مثال

```
Sql>select deptno , Avg(age) From emp
```

```
2      Group by deptno;
```

في المثال السابق قمنا بطلب متوسط حساب متوسط أعمار الموظفين الموجودين في كل دائرة على حد

مثال

```
Sql>select deptno , Avg(age) from emp
```

```
2      where city = 'Riyadh' Group by deptno ;
```

في هذا المثال قمنا باستخدام دالة التجميع مع تحديد شرط معين على نفس جملة الاستعلام فماذا لو أردنا أن نقوم بفرض شرط معين على نفس دالة المجموع

مثال

```
Sql>Select job , Avg(sal) from emp Group By job Having Avg(sal) > 12000;
```

قمنا بالمثال السابق بفرض شرط على دالة التجميع وهذا المثال يقوم بطلب المتوسط الحسابي لرواتب كل منصب من مناصب الموظفين بشرط أن يكون المتوسط أكثر اثنتى عشر ألف

دوال GROUP:

- AVG المعدل
- COUNT عدد
 - MAX اكبر قيمة
 - MIN اقل قيمة
 - STDDEV
 - SUM المجموع
 - VARIANCE

البنية:

```
SELECT [column,] group_function(column), ...  
FROM table  
[WHERE condition]  
[GROUP BY column]  
[ORDER BY column];
```

مثال:

```
Sq|>SELECT AVG(salary), MAX(salary),  
2      MIN(salary), SUM(salary)  
3      FROM employees  
4      WHERE job_id LIKE '%REP%';
```

عملية GROUP BY: هي تقسيم صفوف الجدول الى اصغر مجموعات .

البنية:

```
SELECT column, group_function(column)  
FROM table  
[WHERE condition]  
[GROUP BY group_by_expression]  
[ORDER BY column];
```

مثال:

```
Sq|>SELECT department_id, AVG(salary)  
2      FROM employees  
3      GROUP BY department_id;
```

الخرج :

| DEPARTMENT_ID | AVG(SALARY) |
|---------------|-------------|
| 10 | 4400 |
| 20 | 9600 |
| 50 | 3600 |
| 60 | 6400 |
| 80 | 10033.3333 |
| 90 | 19333.3333 |
| 110 | 10150 |
| | 7000 |

مثال اخر:

```
Sql>SELECT AVG(salary)
2      FROM employees
3      GROUP BY department_id;
```

مثال اخر لاکثر من حقل:

```
Sql>SELECT department_id dept_id, job_id, SUM(salary)
2      FROM employees
3      GROUP BY department_id, job_id;
```

العملية **HAVING**: عملها بدلا من WHERE

مثال

```
Sql>SELECT job_id, SUM(salary) PAYROLL
2      FROM employees
3      WHERE job_id NOT LIKE '%REP%'
4      GROUP BY job_id
5      HAVING SUM(salary) > 13000
6      ORDER BY SUM(salary);
```

تشغيل اسم التقرير في البداية:

```
Sql>TTITLE 'Salary|Report';
```

تشغيل اسم التقرير في النهاية:

```
Sql>BTITLE 'Confidential'
```

عمليات الإضافة

Insert

في حال قمنا ببناء جدول ما فيجب تغذيته بمعطيات معينة لكي نقوم بالاستفادة من قاعدة البيانات لدينا وإجراء العمليات على هذه البيانات والتي تم إعطائه للقاعدة من قبل مدخلي البيانات ولذلك من الواجب تعلم كيفية استخدام لغة التعامل مع البيانات في الإضافة

البنية

Insert Into (القيم) values (أسماء الحقول) اسم الجدول ;

مثال

```
Sql>Insert into emp ( ename, job , city , age , deptno)
```

```
2      Values ('Mohammed','Manager','Riyadh', 23 , 3);
```

قمنا بالمثال السابق بإضافة سجل جديد لموظف جديد مع كامل بياناته ماذا لو أردنا أن نقوم بإضافة بيانات داخل جدول موجودة مسبقا في جدول آخر

مثال

```
Sql>Insert into emp1(ename,job,city,age,deptno)
```

```
2      Select ename,job,city,age,deptno from emp
3      where ename = 'Mohammed';
```

في المثال السابق قمنا بإضافة سجل جديد في جدول آخر موجودة بياناته في جدول ثاني

مثال

```
Sql>Insert Into City(city) Select city from emp;
```

في المثال السابق قمنا بوضع أسماء المدن التي لدينا من جدول الموظفين إلى جدول آخر أنشئناها اسمه جدول المدن

وبذلك نكون قد وضحنا كيفية عملية إدخال البيانات في جدول ما باستخدام الجمل المتداخلة

عمليات التعديل

Update

في حال كان لدينا بيانات في جدول ما وأردنا تعديل بعض أو كل السجلات الموجودة فنستخدم هذه العملية من عمليات لغة التعامل مع البيانات

البنية

Update اسم الجدول Set column=new_value[(column=value..)] WHERE condition;

مثال

Sql>Update emp

```
2 Set city = 'Riyadh'
3 where dept = 1;
```

في المثال السابق قمنا بتعديل بيانات جميع الموظفين في دائرة معينة إلى وضع المدينة التي هم بها إلى الرياض

مثال

Sql>Update emp Set sal = (sal *20)/100

في المثال السابق قمنا بتعديل جميع البيانات الخاصة بالموظفين بإضافة ٢٠ بالمائة من أصل الراتب إلى رواتبهم

مثال

Sql>Update emp Set Comm = (select comm from emp where ename = 'Mohammed');

بهذا المثال قمنا بتعديل بيانات جميع البيانات بجعل العمولة الخاصة بهم تساوي نفس عمولة الموظف الذي أسمه محمد وبهذا المثال نكون قد بينا كيفية استخدام جمل الاستعلام المتداخلة في جملة التعديل

التعديل بدلالة جدول اخر:

Sql>update emp

```
2 set (job,deptno)=(select job,deptno
3 from emp)
4 where empno=7499) رقم الموظف هذا يتعدل(الرقم الملغى)
5 where empno=7699 ; الرقم الذي سيتبدل بدلاً من الاول
```

التوضيح:

في مثالنا السابق سيتم تعديل الرقم عند الشرط الاول الذي سيحل مكانه الرقم الموجود في الشرط الثاني المؤشر

عمليات الحذف

Delete

نحتاج أحيانا إلى حذف سجل أو سجلات من قاعدة البيانات وللقيام بذلك يجب أن نقوم باستخدام جملة لغة استعلام خاصة بالحذف

البيئة

Delete From اسم الجدول ;

مثال

Sql>Delete From emp;

يقوم المثال السابق بحذف جميع الصفوف الموجودة في جدول الموظفين

مثال

Sql>Delete From emp Where ename = 'Mohammed':

قمنا بحذف السجل الخاص بالموظف الذي اسمه محمد وبذلك نكون حددنا شرط معين لنقوم بحذف السجل على أساسه

مثال

Sql>Delete From emp Where Comm = (Select comm From emp Where sal = 3000);

قمنا في المثال السابق بحذف سجلات الموظفين الذين عمولاتهم = عمولات الموظف الذي راتبه ثلاثة آلاف

المثال السابق مثال غير عملي ولكن بينا به كيفية تداخل جمل الاستعلام لحذف سجلات على أساس جملة استعلام معينة

الاستعلامات المتداخلة

يمكن تداخل أكثر من جملة استعلام بين بعضها البعض وتسمى هذه الطريقة أحيانا بالاستعلامات الفرعية وتفيد هذه الطريقة في الحصول على استعلام يعتمد في نتائجه على استعلام آخر

مثال

```
Sql>select ename , job , city From emp
```

```
2      where sal = (select max(sal) from emp) ;
```

والمثال السابق نتائجه هي نفس نتائج المثال التالي

```
Sql>select ename , job , city , max(sal) From emp;
```

ولكن تم كتابة المثال السابق لكي يتم تعريف كيفية استخدام الاستعلامات المتداخلة فيما بينها للحصول على جملة استعلام قوية ومحددة كما ويمكنك أن تقوم ببناء أكثر من جملتين استعلام متداخلتين فيما بينهما للحصول على معلومات أكثر ويجب الأخذ في الاعتبار في أن التنفيذ في هذه الحالة يتم في أقل جمل الاستعلام رتبة ثم يتم تنفيذ الأعلى في ترتيب تصاعدي

مثال

```
select (1)
```

```
(Select (2)
```

```
(Select (3))
```

بالمثال السابق سوف يتم تنفيذ رقم ٣ ثم ٢ ثم ١ بحيث أن نتائج ٢ تعتمد على ٣ ونتائج ١ تعتمد على ٢

مثال

```
Sql>Select ename , sal , job From emp
```

```
2      Where deptno =(Select deptno from emp
3      where City=(select City From emp
4      Where City = 'Riyadh'));
```

المثال السابق غير عملي ولكنه يقوم ببيان المقصود من تداخل أكثر من جملة استعلام فيما بينهما

مثال

```
Sql>Select ename , job ,sal From emp
```

```
2      Where Sal = (Select max(sal) from emp
3      where job = 'salesman');
```

في المثال السابق قمنا بطلب بيانات أعلى راتب من رواتب الموظفين والذي يعمل في قسم المبيعات

دالة الاتحاد Union

تستخدم هذه الدالة لعمل استعلام متداخل على أكثر من جملة استعلام في نفس الوقت دون تداخل

مثال

```
Sql>Select ename ,sal,job,septno from emp
2      where Sal in(Select sal from emp
3      where ename = 'Mohammed' or ename = 'Khalid');
```

في المثال السابق قمنا بطلب بيانات الموظفين بحيث أن يكون راتبهم هو مثل راتب الموظف خالد والموظف محمد وهذه تعتبر أحد الدوال والتي يمكن تحديد عدد من القيم للحصول على مثلها وهي (In 1,2,3,4,5) ولاستخدام جملة الاتحاد سوف نقوم بحل المثال السابق ولكن باستخدام دالة الاتحاد

مثال

```
Sql>Select ename ,sal,job,septno from emp
2      where Sal in (
3      select sal from emp
4      where ename = 'Mohammed' Union
5      select sal from emp
6      where ename = 'Khalid');
```

وبهذا قمنا بعمل جملتي استعلام كل على حدى وقمنا بجعل الجملة الأصلية ببناء قيمتها على أحد القيم الناتجة من أحد جملتي الاستعلام الآخرين

| | | |
|---------------|--|---------------|
| ORACLE | المحاضرة الثالثة | ORACLE |
| | إنشاء والتعامل مع الفهارس Index | |

إنشاء فهرس

الفهرس يستعمل لتسريع عمليات البحث إذ انه يستعمل عند وجود كم هائل من البيانات

| | |
|--------------|-------------|
| CREATE INDEX | إنشاء منظار |
|--------------|-------------|

التوصيف

CREATE INDEX IN **اسم الفهرس** ON **اسم الجدول** (**اسم الحقل**)

مثال

SQL> create index **aaa** on phone2(person_name);

توضيح

لقد قمنا بإنشاء فهرس على اسم الموظف من جدول الموظفين

لإنشاء فهرس وحيد أي لا يقبل قيم متكررة

SQL> create **unique** index **aaa** on phone2(person_name);

لحذف الفهرس

SQL> drop index **aaa** ;

| | | |
|--------|---------------------------------------|--------|
| ORACLE | المحاضرة الرابعة إنشاء العروض View | ORACLE |
|--------|---------------------------------------|--------|

إنشاء العروض أو المناظير

المنظار هو عبارة عن تقرير معد مسبق ومخزن في الجهاز لطلب بيانات معينة وذلك عن طريق جملة استعلام أو عدة جمل استعلام متداخلة كما يمكن أن نقول عنه أنه سكر بيت ولكنه يقوم باستدعاء البيانات عند استدعائه ولا يقوم بحفظها كما تفعل اللقطات الثابتة المأخوذة من الجدول والتي لا يتم حصول أي تعديلات عليها عند حدوث أي تغيير على النسخة الأصلية ولكن العرض يقوم بالتغير، بمعنى أعم أنه عبارة عن إنشاء جدول VIEW تنشأ بدلالة الجداول الأساسية ويمكن إنشاءه من حقل أو حقلين أو أكثر وعمل له منطقة عمل للتجربة بعيداً عن الجدول الأصلي..

| | |
|-------------|-------------|
| CREATE VIEW | إنشاء منظار |
|-------------|-------------|

الوصف

CREATE VIEW اسم المنظار AS SELECT (جملة استعلام)

مثال

Sql>CREATE VIEW SSS AS SELECT * FROM EMP;

توضيح

لقد تم إنشاء منظار يحتوي على جملة استعلام تقوم بإظهار كافة البيانات من الجدول الثاني

عمل آخر للـ VIEW:

عمل جدول بإعطاء اسم الأعمدة أسماء يفهمها المستخدم غير الأسماء التي يوضعها المبرمج لكي يسهل على المستخدم طباعتها

البنية:

Create VIEW view-name AS SELECT { اسم العمود للظهور وقت { اسم العمود لوجهة المبرمج } { اسم العمود للطباعة } {2 } , FROM table-name WHERE condition ;


```
Sql>CREATE VIEW
2 AS SELECT employee_id ID_NUMBER, last_name NAME,
3 salary*12 ANN_SALARY
4 FROM employees
5 WHERE department_id = 50;
```

المثال السابق لإنشاء جدول وهمي لإعطاء رقم واسم الموظف وراتبه *١٢ من جدول الموظفين عند القسم ٥٠

لاستدعاء المنظار

يمكنك التعامل معه كما لو كان جدول عن طريق جمل الاستعلام

```
Sql>SELECT * FROM salvu50 ;
```

وعندما استدعينا نفس المنظار السابق سيكون الناتج قد تغير في اسماء الحقول المؤشر عليها كالتالي...

| ID_NUMBER | NAME | ANN_SALARY |
|-----------|---------|------------|
| 124 | Mourgos | 69600 |
| 141 | Rajs | 42000 |
| 142 | Davies | 37200 |
| 143 | Matos | 31200 |
| 144 | Vargas | 30000 |

لحذف المنظار

```
DROP view salvu50;
```

المحاضرة الخامسة

بعض جمل الاستعلام الخاصة

هناك بعض جمل الاستعلام الخاصة بمعرفة بعض الأمور المتعلقة بالجدول ولمستخدمين والتي تفيد كثير مديري قاعدة البيانات ،، وهذه صيغ تستعمل للاستعلام عن معلومات خاصة عن المستخدمين

| | |
|---|--|
| SELECT * FROM USER_TABLES; | تستخدم لمعرفة المعلومات الخاصة عن الجداول الخاصة بكل مستخدم |
| SELECT * FROM DBA_USERS; | تستخدم لمعرفة جميع المعلومات عن المستخدمين ويستخدمها مدير النظام فقط |
| SELECT * FROM ALL_USERS; | تستخدم لمعرفة أسماء المستخدمين الموجودين وأرقامهم وتواريخ إنشائهم |
| SELECT * FROM USER_INDEXES; | تستخدم لمعرفة المعلومات الخاصة عن الفهارس الموجودة |
| SELECT * FROM USER_VIEWS; | تستخدم لمعرفة المعلومات الخاصة عن المناظير الموجودة |
| SELECT granted_role FROM user_role_privs | هذه الجملة تقوم بإعطائك السماحيات الخاصة بالمستخدم الحالي |
| SELECT COUNT(*) FROM SYSTEM_PRIVILEGE_MAP | تقوم هذه الجملة بإعلامك بعدد الصفوف أو الامتيازات الموجودة في النظام |
| SELECT USER FROM DUAL | تقوم هذه الجملة بإعطائك اسم المستخدم الحالي |
| SELECT owner, table_name FROM all_tables WHERE owner NOT IN('SYS','SYSTEM') | تقوم هذه الجملة بإعلامك بجميع أسماء الجداول الخاصة بالمستخدمين |
| SELECT synonym_name FROM all_synonyms | تقوم هذه الجملة بإخبارك بجميع أسماء المرادفات الموجودة في النظام |
| SELECT * from V\$DBFILE | تقوم هذه الجملة بإخبارك عن أسماء ومسار ملفات البيانات الخاصة بقاعدة البيانات |
| select VALUE from V\$PARAMETER Where Name = 'control_files' | تقوم هذه الجملة بإخبارك عن اسم ومسار ملف التحكم الخاص بالنظام |
| Select * From V\$DATABASE | تقوم هذه الجملة بإخبارك بمعلومات حول اسم قاعدة البيانات وحالة ملف التسجيل وتاريخ الإنشاء |
| Select * From V\$BACKUP | تقوم هذه الجملة بإخبارك عن بيانات النسخ الاحتياطي |
| Select * From V\$RECOVERY_FILE | تقوم هذه الجملة بإخبارك عن بيانات الاسترداد |
| Select * from USER_FREE_SPACE | تقوم هذه الجملة بإخبارك عن المساحة المتبقية للمستخدم |
| select * from ALL_TABLES | تقوم هذه الجملة بإعلامك عن معلومات جميع الجداول كالأسماء واسم الجدول ومساحته |
| select TABLE_NAME from ALL_TABLES | تقوم هذه الجملة بإخبارك عن أسماء جميع الجداول الموجودة |
| Select * from ALL_USERS | تقوم هذه الجملة بإخبارك بمعلومات حول المستخدمين وتاريخ إنشائهم |
| Select * from ALL_VIEWS | تقوم هذه الجملة بإخبارك عن معلومات كاملة عن المناظير أو العروض |
| Select VIEW_NAME from ALL_VIEWS | تقوم هذه الجملة بإخبارك عن أسماء جميع العروض النظام الموجودة في |
| select SQL_TEXT from V\$SQL | تقوم هذه الجملة بإعلامك بجميع جمل الاستعلام القاعدة المستخدمة والمنفذة في |

الجميل السابقة يمكن من خلالها تركيب جمل لمعرفة أو للحصول على استعلامات محددة

محاضرة اضافية

مصطلحات في عالم الكمبيوتر

هذه بعض المصطلحات والاختصارات التي تستعمل في عالم الكمبيوتر وأوراكل والتي لاغنى عنها في عالم الكمبيوتر وترجمتها بالعربية والانجليزية وقد توجد بعض الأخطاء في الترجمة ولكن هذا جهد لذي يتم توفير أكثر المصطلحات الخاصة مع تقرب ترجمتها إلى العربي

| مصطلحات خاصة بأوراكل وبعض لغات البرمجة | | |
|--|--|---|
| المصطلح | الترجمة الانكليزية | الشرح والمعنى العربي |
| SQL | Structured Query Language | إنشاء الاستفسارات لغة |
| DDL | Data Definition Language | توصيف البيانات لغة |
| DML | Data Manipulation Language | التعامل مع البيانات لغى |
| DB | DataBase | بيانات قاعدة |
| DBA | DataBase Administrator | قواعد البيانات مدير |
| DBMS | DataBase Management System | نظم قواعد البيانات ادرارة |
| ODBC | Open DataBase Connectivity | قواعد البيانات النشطة ربط |
| VLDB | Very Large DataBase | البيانات الكبيرة قواعد |
| RDMD | Relational DataBase Management System | البيانات العلائقية قواعد |
| ORDBMS | Object-Relational DataBase Management System | قواعد بيانات علائقية غرضية التوجه إدارة |
| LOB | Large Object Binary | كبيرة جدا بيانات |
| GUI | Graphic User Interface | المستخدم الرسومية واجهة |
| PL | Procedure Language | الأجراءات لغة |
| OLAP | Online Analytical Processing | التحليلية المباشرة المعالجة |
| SGA | System Global Area | النظام العامة منطقة |
| PGA | Program Global Area | البرامج العامة منطقة |
| SID | System Identifier | النظام معرف |
| ADO | ActiveX Data Objects | أكتيف أكس للوصول الى البيانات كائنات |
| DAO | Data Access Object | الوصول الى البيانات كائنات |
| DDE | DynamicData Exchange | التبادل الديناميكي كائنات |
| VBA | Visual Basic for Application | فيجوال بيسك للتطبيقات لغة |
| RDO | Remote Data Objects | التحكم بالبيانات البعيدة كائنات |
| VBS | Visual Basic Scribt | فيجوال بيسك الخاصة بالانترنت لغة |
| OTN | Oracle Technology Network | اوراكل التكنولوجيا شبكة |
| WYSIWYG | What You See Is What You Get | هو ماتحصل عليه ماتراه |

| مصطلحات خاصة في عالم الإنترنت | | |
|-------------------------------|-----------------------|----------------------------|
| المصطلح | الترجمة الانكليزية | الشرح والمعنى العربي |
| WWW | World Wide Web | العنكبوتية العالمية الشبكة |
| .com | Commercial Businesses | تجاري موقع |
| .edu | Higher Education | للتعليم العالي موقع |
| .org | Organization | منظمات أو هيئات مواقع |
| .gov | Government | حكومية مواقع |
| .net | Network | للشبكات مواقع |

| | | |
|------------|---|---|
| .mil | Military | عسكرية مواقع |
| HTTP | Hypertext Transfer Protocol | النص لغة نقل |
| HTML | Hypertext Markup Language | إعداد النص لغة |
| DHTML | Dynamic HTML | HTML الديناميكية |
| FTP | File Transefer Protocol | الملفات لغة نقل |
| IP Address | Internet Protocol Address | تعريف الإنترنت عنوان |
| ISP | Internet Server Provider | خدمة الإنترنت مقدمة |
| W3C | World Wide Web Consortium | تحديد معايير لغة انشاء الصفحات جمعية |
| CGI | Common Gateway Interface | المباشرة الطرق |
| P.P.P | Point-toPoint Protocol | من نقطة إلى نقطة بروتوكول |
| TCP/IP | Transfer Control Protocol / Internet Protocol | تحكم النقل للإنترنت بروتوكول |
| SLIP | Serisl Line Interface Protocol | متوفر حاليا غير |
| CSS1 | Cascading Style Sheets 1 | متوفر حاليا غير |
| SSI | Server Side Include | متوفر حاليا غير |
| DOM | Document Object Modal | متوفر حاليا غير |
| IIS | Internet Information Server | معلومات الإنترنت خادم |
| PWS | Personal Web Server | شبكة شخصي خادم |
| IPP | Internet Presence Provider | متوفر حاليا غير |
| PGP | Pretty Good Privacy | أمن خادم |
| XML | eXtnsible Makup Language | التوصيف القابلة للتمدد - شبيه لغة HTML |
| ASP | Active Server Page | الخادم النشطة صفحة |
| SSL | Secure Socket Layer | أو البروتوكول ذو الطبقة الأمانة المقبس |
| SET | Secure Elecronic Transaction | الإلكتروني الأمن التعامل |
| SMTP | Simple Mail Transfer Protocol | نقل البريد البسيط بروتوكول |
| NNTP | Network News Transfer Protocol | متوفر حاليا غير |
| FAQ | Frequently Asked Questions | المتكررة الاسئلة |
| ISDN | Integrated Services Digital Network | الرقمية للخدمات بسرعات ٦٤ و ١٢٨ الشبكة كيلوبايت |
| NNTP | Network News Transport Protocol | خدمات النقاش بروتوكول |
| POP | Post Office Protocol | البريد الإلكتروني بروتوكول |
| SLIP | Serial Line Internet Protocol | البروتوكول التسلسلي للاتصال بالإنترنت |
| URL | Uniform Resource Locator | وصلة انترنت اختصار |
| IRC | Internet Relay Chat | المحادثة عبر الإنترنت خدمة |
| TFTP | Trivial File Transfer FTP | متوفر حاليا غير |
| PHP | Hypertext Preprocessor | برمجة شبيهة ب لغة HTML |

منظمات خاصة بعامل الكمبيوتر والإنترنت

| المصطلح | الانكليزية الترجمة | والمعنى العربي الشرح |
|---------|--|---------------------------------------|
| ARPAnet | Advanced Research Projects Agency | البحوث المتقدمة شبكة |
| BBS | Bulletin Board System | لوحات النقاش نظام |
| CERT | Computer Emergency Response Team | طوارئ أمن الشبكة مركز |
| CERN | European Laboratory for Particle Physics | WWW المختبر الذي ابدعة فيه الشبكة اسم |
| CIX | Commercial Internet Exchange | وضع قواعد الانترنت منظمة |
| DDN | Defense Data Network | معلومات قطاعات الدفاع شبكة |

| | | |
|------|---------------------------------|---|
| DOD | Department of Defense | الدفاع الأمريكية التي لاجلها انشأت الانترنت وكالة |
| EFF | Electronic Frontier Foundation | الحدود الالكترونية مؤسسة |
| IAB | Internet Architecture Board | هندسة الانترنت هيئة |
| IETF | Internet Engineering Task Force | مهندسي مطوري اعمال الانترنت هيئة |
| ISOC | The Internet Society | الانترنت جمعية |
| NIC | Network Information Center | معلومات الشبكة مركز |
| NIC | Network Interface Card | اتصال الشبكة بطاقة |
| VPN | Virtual Private Network | الالكترونية خاصة شبكة |
| ACH | Automated Clearing House | ادارة الاموال شبكة |

الكفية مصطلحات خاصة بالاجهزة بالم والاجهزة Palam Windows CE WAP

| المصطلح | الترجمة الانكليزية | الشرح والمعنى العربي |
|------------|-----------------------------------|---|
| Palam | - | تشغيل خاص بالاجهزة الكفية نظام |
| Windows CE | - | تشغيل خاص بالاجهزة الكفيه شبيه بويندوز نظام |
| WAP | Wireless Application Protocol | النقل اللاسلكي لانترنت مثل الهواتف بروتوكول التقالة |
| HDML | Hand-Held Device Mark-up Language | توصيف النص الخاصة بالاجهزة لغة HTML الكفية شبيهة |
| WCA | Web Clipping Application | لتحويل الصفحات لاجهزة كفية برنامج |
| POSE | Palam OS Emuator | |
| PQA | Palam Query Application | |
| GPS | Global Positioning System | الواجهات العالمي نظام |
| WML | Wireless Mark-up Language | توصيف النص اللاسلكية لغة |
| DTDs | Document Type Definition's | تعريف النص لغة |

مصطلحات خاصة في أجهزة الحاسب

| المصطلح | الترجمة الانكليزية | الشرح والمعنى العربي |
|---------|-------------------------|-------------------------------------|
| PC | Personal Computer | شخصي حاسب |
| RAM | Randome Access Memory | العشوائية الذاكرة |
| ROM | Read Only Memory | القراءة فقط ذاكرة |
| CPU | Central Processing Unit | المعالجة المركزية وحدة |
| CD | Compact Disc | مضغوط محرك قرص |
| CD-R | CD Write | للقرص المضغوط محرك نسخ |
| CD-Re | CD Rewrite | معيد الكتابة على القرص المضغوط محرك |
| HD | Hard Disc | الصلب القرص |
| MB | Mother Board | الأم اللوحة |
| KB | Key Board | المفاتيح لوحة |
| Kbps | Kelobits Per Second | لكل ثانية كيلوبايت |

مصطلحات خاصة في علوم الحاسب

| المصطلح | الترجمة الانكليزية | الشرح والمعنى العربي |
|---------|---------------------------------------|--|
| ANSI | American National Standards Institute | الأمريكية الدولية للمقاييس الجمعية |
| UCS | Unicode World Wide Character Standard | مقاييس أحرف التصفح جمعية |
| RGB | Red & Green & Blue | القياسية الأحمر والأخضر و الأزرق الألوان |
| Bit | Bit | وحدات القياس أصغر |
| Byte | 8 Bit = 1 Character | 8 بت |

| | | |
|--------|--|--------------------------------------|
| KB | 1024 Byte = KeloByte | 1024 بايت |
| MB | 1024 KB Miga Byte | بايت ١٠٢٤ كيلو |
| GB | 1024 MB Giga Byte | 1024 ميجابايت |
| TB | 1024 GB Titra Byte | ١٠٢٤ تيترا بايت |
| VRML | Virtual Reality Markup Language | برمجة لغة |
| API | Application Programming Interface | مبرمج التطبيقات واجهة |
| CLSID | Class Identifier | الصفوف معرف |
| DCOM | Distributed Component Object Model | متوفر حاليا غير |
| IMS | Information Management System | نظام المعلومات مدير |
| IIS | Internet Information Server | بيانات الإنترنت خادم |
| CICS | Customer Information Control System | إدارة بيانات الزبائن نظام |
| COMTI | COM Transaction Integrator | متوفر حاليا غير |
| DAO | Data Access Objects | الوصول للبيانات كائنات |
| DBCS | Double Byte Characters Set | متوفر حاليا غير |
| DDE | Dynamic Data Exchange | تبادل ديناميكية بيانات |
| DLL | Dynamic Link Library | الربط الديناميكية مكتبات |
| EBCDIC | Extended Binary Coded Decimal Interchange Code | متوفر حاليا غير |
| ERA | Entity Relationship Analysis | العلاقات الشخصية تحليل |
| GUID | Globally Unique Identifier | العالمي الموحد التعريف |
| IPC | InterProcess Communication | متوفر حاليا غير |
| ISAPI | Internet Server API | مبرمج التطبيقات لخادم الإنترنت واجهة |
| MDAC | Microsoft Data Access Components | الوصول لبيانات ميكروسوفت مكونات |
| MDI | Multiple Document Interface | متعددة الوثائق واجهة |
| SDI | Single Document Interface | ذات وجه واحد واجهة |
| MSF | Microsoft Solution Framework | حلول ميكروسوفت إطار |
| MSMQ | Microsoft Message Queue Server | صفوف الرسائل لميكروسوفت خادم |
| MTS | Microsoft Transaction Server | أعمال مايكروسوفت خادم |
| OLE | Object Linking and Embedding | وتضمين الكائنات ربط |
| SMS | System Management Server | مدير النظام خادم |
| SNA | System Network Architecture | نظم الإنترنت هندسة |
| UDA | Universal Data Access | العالمي للبيانات الوصول |
| ACID | Consistency Isolation and Durability | والعزل والأمان التماسك |
| ADO | ActiveX Data Objects | الوصول بأكتف أكس كائنات |
| UDT | Uniform Data Transfer | متوفر حاليا غير |

COM & COM & COM

| المصطلح | الترجمة الأنكليزية | الشرح والمعنى العربي |
|---------------|------------------------|-------------------------|
| COM DOS | Command | أمر |
| .COM Internet | Commercial | شركة |
| COM ActiveX | Component Object Model | البيانات بالبرامج تضمين |

معاني إمتداد بعض الملفات

| المصطلح | الترجمة الأنكليزية | الشرح والمعنى العربي |
|---------|--------------------|----------------------|
| EXE | Executable File | تنفيذي ملف |
| BAT | Batch File | دفعي ملف |
| COM | Command File | أوامر ملف |

| | | |
|-------------|--|-------------------------|
| TXT | Text File | نصي ملف |
| SYS | System File | نظام ملف |
| WAV | Wave File | صوتي ملف |
| MIDI | Music Instrument Digital Interface File | وسائط صوتي ملف |
| MID | Music Instrument Digital File | وسائط صوتي ملف |
| QT | Quick Time File | فيديو ملف |
| BMP | Bitmap File | صورة ملف |
| EMF | Enhanced Meta File | صورة ملف |
| WMF | Windows Meta File | صورة ملف |
| GIF | Graphic Interchange Format File | صورة ملف |
| AVI | Audio Video Interleave File | فيديو ملف |
| DOC | Document File | ورد وثيقة |
| MDB | Microsoft Database File | قواعد بيانات ملف |
| MPEG | Moving Pictures Experts Group File | فيديو ملف |
| PIC | PC Paint | صورة ملف |
| TRC | Audio Trac File | قرص ملف صوت |
| JPEG | Joint Photographic Experts Group File | صورة ملف |
| PNG | Portable Network Graphics File | صورة ملف |
| TIFF | Tagged Image File Format File | صورة ملف |
| PDF | Portable Document Format File | صورة ملف |
| MP3 | MPEG-Layer3audio File | صورة ملف |

أوامر إضافية في حساب المستخدم الخاص:

بعد الدخول إلى محرر الاستعلامات في حساب المستخدم الخاص بك تفحص القوائم الخاصة بالمحرر

| القائمة ملف File | |
|------------------------|---|
| Open | فتح ملف تم حفظه مسبقا |
| Save | حفظ |
| Create | انشاء ملف جديد |
| Replace | استبدال ملف موجود |
| Append | إضافة إلى ملف موجود |
| Save As | حفظ باسم |
| Spool | ملف التخزين يقوم بتخزين جميع ماتقوم به من عمليات ونتائج |
| Spool File | فتح وتحليل ملف التخزين |
| Spool Off | اغلاق ملف التخزين |
| Run | تنفيذ الامر المكتوب حاليا أو تم كتابته آخر مرة |
| Cancel | الغاء الأمر الذي يتم تنفيذه حاليا |
| Exit | الخروج من البرنامج |
| القائمة تحرير Edit | |
| Copy | نسخ |
| Paste | لصق |
| Clear | تنظيف الشاشة |
| Editor | خيارات المحرر |
| Invoke Editor | يقوم بفتح المفكرة على آخر أمر تم فتحه لكي يتم التعديل به وحفظه |
| Define Editor | يقوم بتعريف محرر غير المفكرة في استخداماته مع محرر الاستعلامات |
| القائمة بحث Serch | |
| Find | بحث عن كلمة محددة |
| Find Next | بحث عن التالي |
| القائمة خيارات Options | |
| Environment | يقوم بفتح الخصائص الخاصة بالمحرر كحجم الخازن المؤقت وغيرها من الثوابت التي تستخدم في المحرر |

بعد أن تعرفنا على قائمة التحرير يجب أن نتعرف على بعض الأوامر الأخرى التي يمكننا من زيادة التعامل مع المحرر فمثلا هناك أوامر مختصرة للتي توجد في القائمة وهناك أوامر أخرى من الممكن اعتبارها جمل استعمال ومن الممكن اعتبارها أيضا أوامر المحرر ولكن في جميع الحالات فأن مكان شرحها في هذا القسم بسبب علاقتها الوطيدة به

| الامر | التوضيح |
|------------------|--|
| Ed | استعمال تعمل عمل Invoke يستعمل لفتح المفكرة لتعديل آخر جملة |
| R | تستعمل لإعادة تنفيذ آخر جملة تعمل عمل RUN |
| Clear SCR | تستعمل لتنظيف الشاشة من المحتويات مثل CLS |
| Save WWW | تستعمل لحفظ جمل استعمال باسم معين إلى حين استدعائها |
| Get WWW | تستعمل لتعديل جملة تم حفظها من قبل |
| Start WWW | تستعمل لتنفيذ جملة محفوظة من قبل |
| Save WWW Replace | تستعمل لتغير اسم حفظ آخر باسم جديد |
| Roll Back | للتراجع عن جميع العمليات السابقة ولا تشمل أوامر تعريف البيانات والتراجع يكون من آخر جملة تثبيت |
| Commit | تستعمل لتثبيت البيانات تثبيت نهائي |
| Desc TName | لبيان بنية الجدول المحدد الشكل الفيزيائي للجدول |
| Disc | لقطع اتصال المستخدم في قاعدة البيانات |
| Connect | لاتصال مستخدم جديد في قاعدة البيانات |
| Spool FName | لفتح ملف التخزين المؤقت |
| Spool Off | لاغلاق ملف التخزين المؤقت |

نعم بحمد الله

تابعوا الاصدار الثاني ملخص PL/SQL

المصادر:

*كتاب تعلم الاوراكل بسهولة - - الكاتب الاسيف

*مقدمة الى الاوراكل SQL 9i - للمؤلفان:

Priya Nathan & Nancy Greenberg

*تعليم الاوراكل للعرب - بدوي سوفت

*كتاب من منتديات الفريق العربي للبرمجة

مع تحياتي للجميع بالتوفيق والنجاح

الفهرس

| الصفحة | المواضيع |
|--------|---|
| ٢ | المقدمة |
| ٣ | عزيزي القارئ |
| ٤ | الاهداء |
| ٥ | المحاضرة الاولى: لغة تعريف البيانات DDL |
| ٥ | انشاء جدول |
| ٦ | حذف جدول |
| ٦ | اضافة حقل في جدول منشأ |
| ٧ | تعديل بيانات حقول |
| ٨ | المحددات |
| ١٣ | انشاء مستخدم جديد |
| ١٤ | المحاضرة الثانية: لغة التعامل مع البيانات DML |
| ١٤ | SELECT |
| ١٩ | ORDER BY |
| ٢٠ | وظائف خاصة بالحروف |
| ٢١ | عمليات ربط الجدول "خارجي، متساوي، مع نفسه" |
| ٢٣ | دالة التجميع GROUP BY & HAVING |
| ٢٦ | الاضافة INSERT |
| ٢٧ | التعديل UPDATE |
| ٢٨ | الحذف DELETE |
| ٢٩ | الاستعلامات المتداخلة |
| ٣١ | المحاضرة الثالثة: انشاء الفهارس INDEX |
| ٣٢ | المحاضرة الرابعة: انشاء العروض VIEW |
| ٣٤ | المحاضرة الخامسة: بعض جمل الاستعلام الخاصة |
| ٣٥ | مصطلحات في عالم الكمبيوتر |
| ٤٠ | اوامر اضافية في حساب مستخدم خاص |
| ٤١ | المصادر |
| ٤٤ | السيرة الذاتية لنجم الويب |
| ٤٤ | الخاتمة |

التعريف بـ WebStar :

الإسم: علي احمد علي قاسم

الجمهورية اليمنية

موايد محافظة الحديدة مديرية الخوخة

العمر: ٢٠ سنة

طالب جامعي في كلية علوم وهندسة الحاسوب

المؤهلات: برجة وصيانة الحاسوب مصمم مواقع

الخاتمة:

**أتمنى من الله أن اكون قد وفقت في عملي هذا واتمنى من الله أيضا أن ينال هذا
البحث إعجابكم ورضاكم وفي الأخير .
صلوا على نبي الامه وسيد الخلق**

**محمداً صلى الله عليه وسلم
ورجائي لكم بالدعاء لي ولجميع من صلى على نبينا الصادق الامين
والحمد لله رب العالمين....**



خاص بنجم الويب

كلية علوم وهندسة الحاسوب

مستوى ثاني

جامعة الحديدة