إنشاء وإدارة الجداول (Creating and Managing Tables):

قواعد تسمية الجداول والأعمدة:

1. يجب أن يبدأ اسم الجدول أو اسم العمود بحرف.
2. يجب ألا يتجاوز طول الاسم 30 محرف.
3. يمكن أن يتكون الاسم من المحارف التالية **فقط**: A-Z , a-z , 0-9 , _ , $, #
4. لا يمكن تسمية جدولين أو عنصرين من عناصر قاعدة البيانات يملكهما نفس المستخدم بنفس الاسم.
5. لا يمكن استخدام تعليمات SQL المحجوزة كأسماء للجداول أو الأعمدة.
6. الأسماء غير حساسة لحالة الأحرف.

إنشاء جدول:

لإنشاء جدول في قاعدة البيانات لا بد أن نملك السماحية اللازمة لإنشاء جدول وهي تسمى (CREATE TABLE privilege) وكذلك لا بد أن يتوفر لدينا مساحة كافية لتخزين الجدول، إن الشكل العام لتعليمة إنشاء جدول هو:

```
CREATE TABLE [schema.]table
             (column datatype [DEFAULT expr][, ...]);
```

أي يجب أن نعطي اسم للجدول، أسماء للأعمدة، نوع بيانات كل عمود، والحجم المتوقع لبيانات كل عمود.

مثال:

```
CREATE TABLE dept
            (deptno  NUMBER(2),
             dname   VARCHAR2(14),
             loc     VARCHAR2(13));
```

يمكن عرض بنية الجدول عن طريق التعليمة: DESCRIBE dept ;

إن تعليمة إنشاء جدول لها تاثير مباشر على قاعدة البيانات، أي لا تحتاج إلى تثبيت COMMIT أو إلغاء ROLLBACK، بعكس التعليمات الأخرى مثل INSERT, DELETE, UPDATE

ملاحظة: عند استخدام جداول تابعة لمستخدم آخر، يتم استخدام الصيغة التالية:

اسم الجدول.اسم المستخدم

استخدام الكلمة المحجوزة DEFAULT:

تستخدم لإضافة قيمة افتراضية للعمود في حال لم يتم إدخال قيم من قبل المستخدم، يمكن للقيمة الافتراضية أن تكون أي قيمة محرفية، أو تعابير، أو توابع SQL.بينما لا يمكن استخدام اسم عمود آخر كقيمة افتراضية.

يجب أن تتطابق نوع بيانات القيمة الافتراضية مع نوع بيانات العمود.

أنواع بيانات الأعمدة:

Data type	Description
VARCHAR2(*size*)	Variable-length character data (a maximum *size* must be specified: Minimum *size* is 1; maximum *size* is 4000)
CHAR [(*size*)]	Fixed-length character data of length *size* bytes (default and minimum *size* is 1; maximum *size* is 2000)
NUMBER [(*p*,*s*)]	Number having precision *p* and scale *s* (The precision is the total number of decimal digits, and the scale is the number of digits to the right of the decimal point; the precision can range from 1 to 38 and the scale can range from -84 to 127)
DATE	Date and time values to the nearest second between January 1, 4712 B.C., and A.D. December 31, 9999
LONG	Variable-length character data up to 2 gigabytes
CLOB	Character data up to 4 gigabytes
RAW(*size*)	Raw binary data of length *size* (a maximum *size* must be specified. maximum *size* is 2000)
LONG RAW	Raw binary data of variable length up to 2 gigabytes
BLOB	Binary data up to 4 gigabytes
BFILE	Binary data stored in an external file; up to 4 gigabytes
ROWID	Hexadecimal string representing the unique address of a row in its table. This datatype is primarily for values returned by the ROWID pseudocolumn.

إنشاء جدول باستخدام الاستعلام الفرعي:

الشكل العام:

```
CREATE TABLE table
        [(column, column...)]
AS subquery;
```

ملاحظات:

1. بهذه الطريقة يتم إنشاء الجدول وإضافة بيانات إليه وهي البيانات المستعادة من الاستعلام الفرعي.
2. يتضمن تعريف العمود بهذه الطريقة اسم العمود والقيمة الافتراضية لذلك العمود فقط.
3. في حال لم يتم تعريف أعمدة، فإن أسماء أعمدة الجدول الجديد هي نفسها المستخدمة في عبارة الاستعلام الفرعي.
4. أنواع بيانات الأعمدة المستخدمة في عبارة الاستعلام الفرعي، هي نفسها ستكون أنواع بيانات الأعمدة للجدول الجديد.

مثال:

```
CREATE TABLE  dept80
  AS
    SELECT  employee_id, last_name,
            salary*12 ANNSAL,
            hire_date
    FROM    employees
    WHERE   department_id = 80;
```

نستخدم التعليمة DESCRIBE لعرض بنية الجدول الجديد.

نلاحظ أنه عند استخدام تعابير حسابية في الأعمدة المستخدمة في الاستعلام الفرعي كما في SALARY*12 لا بد من إعطائها أسماء مستعارة وإلا سنحصل على رسالة خطأ.

تعديل بنية الجدول:

تستخدم تعليمة ALTER TABLE لإضافة عمود جديد، تعديل عمود موجود، تعريف قيمة افتراضية لعمود جديد، حذف عمود.

الشكل العام لهذه التعليمة:

```
ALTER TABLE table
ADD         (column datatype [DEFAULT expr]
            [, column datatype]...);
```

```
ALTER TABLE table
MODIFY      (column datatype [DEFAULT expr]
            [, column datatype]...);
```

```
ALTER TABLE table
DROP       (column);
```

مثال:

```
ALTER TABLE dept80
ADD         (job_id VARCHAR2(9));
```

نلاحظ أن العمود المضاف سيكون آخر عمود في الجدول.

مثال 2:

```
ALTER TABLE   dept80
MODIFY        (last_name VARCHAR2(30));
```

تستخدم عبارة MODIFY لتعديل نوع بيانات عمود او حجمه أو القيمة الافتراضية.

ملاحظات:

1. يمكن زيادة الدقة في البيانات من نوع Numric أو Character لكن لا يمكن إنقاصها إلا إذا كان العمود يتضمن قيم NULL أو إذا كان الجدول لا يتضمن صفوف.
2. لا يمكن تغيير نوع بيانات عمود إلا إذا كان العمود يتضمن قيم NULL
3. يمكن تغيير نوع البيانات من VARCHAR2 إلى CHAR، أو من CHAR إلى VARCHAR2 فقط في حال كان العمود يتضمن قيم NULL أو في حال لم يتم تغيير الحجم.

مثال3:

```
ALTER TABLE   dept80
DROP  COLUMN   job_id;
```

تستخدم عبارة DROP لحذف أعمدة.

ملاحظات:

1. باستخدام هذه التعليمة يمكن حذف عمود واحد فقط في كل مرة.
2. لا يمكن حذف كامل أعمدة الجدول، وإنما يجب التأكد من بقاء عمود واحد على الأقل بعد استخدام هذه التعليمة.
3. حالما يتم حذف العمود لا يمكن استعادته.

حذف جدول:

باستخدام تعليمة DROP TABLE يمكن حذف الجدول وعندها سيتم حذف كامل بيانات الجدول، وهذه التعليمة لها تأثير مباشر على قاعدة البيانات أي لا يمكن التراجع عنها (ROLLBACK) بعد تنفيذها، كما أنها ليست بحاجة إلى تثبيت (COMMIT).

```
DROP TABLE dept80;
```

هناك تعليمة أخرى لحذف الجدول وهي TRUNCATE TABLE، هذه التعليمة تقوم بحذف كافة أسطر الجدول، كما أنها تحرر المساحة التخزينية التي كان يحتلها ذلك الجدول.

مثال:

```
TRUNCATE TABLE detail_dept;
```

يجب الانتباه إلى أنه لا يمكن استرجاع الأسطر المحذوفة عند استخدام هذه التعليمة.

كذلك يجب الانتباه أنه في التعليمتين السابقتين، لا يمكنك حذف الجدول إلا إذا كنت أنت من أنشئ ذلك الجدول، أو تملك سماحية خاصة تدعى (DELETE TABLE).

القيود CONSTRAINTS:

يتم استخدام ما يسمى بالـ CONSTRAINTS لمنع إدخال البيانات الخاطئة للجدول، وهي مجموعة من القواعد تطبق إما على مستوى العمود، أو على مستوى الجدول، ويتم تعريف هذه القيود عند إنشاء الجدول، كما يمكن تعريفها بعد إنشاء الجدول.

وهذه القيود هي:

Constraint	Description
NOT NULL	Specifies that the column cannot contain a null value
UNIQUE	Specifies a column or combination of columns whose values must be unique for all rows in the table
PRIMARY KEY	Uniquely identifies each row of the table
FOREIGN KEY	Establishes and enforces a foreign key relationship between the column and a column of the referenced table
CHECK	Specifies a condition that must be true

الشكل العام لتعريف القيود:

```
CREATE TABLE [schema.]table
    (column datatype [DEFAULT expr]
    [column_constraint],
    ...
    [table_constraint][,...]);
```

يتم هنا تعريف القيود على مستوى العمود

يتم هنا تعريف القيود على مستوى الجدول

يُعرّف القيد عن طريق استخدام الكلمة المحجوزة CONSTRAINT ثم يتم إعطاء اسم لهذا القيد، ثم نوع القيد ويتم استخدام أحد القيود المذكورة سابقاً، ثم اسم العمود أو الأعمدة التي سيطبق عليها القيد.
كما يمكن تعريفه على مستوى العمود بطريقة أخرى وهي الاكتفاء فقط بذكر نوع القيد المطبق على ذلك العمود، بعد التصريح عن العمود.

الشكل العام لتعريف القيد على مستوى العمود:

```
column [CONSTRAINT constraint_name] constraint_type,
```

والشكل العام لتعريف القيد على مستوى الجدول:

```
column,...
  [CONSTRAINT constraint_name] constraint_type
  (column, ...),
```

يتم تعريف القيد على مستوى الجدول بعد تعريف كامل أعمدة الجدول.

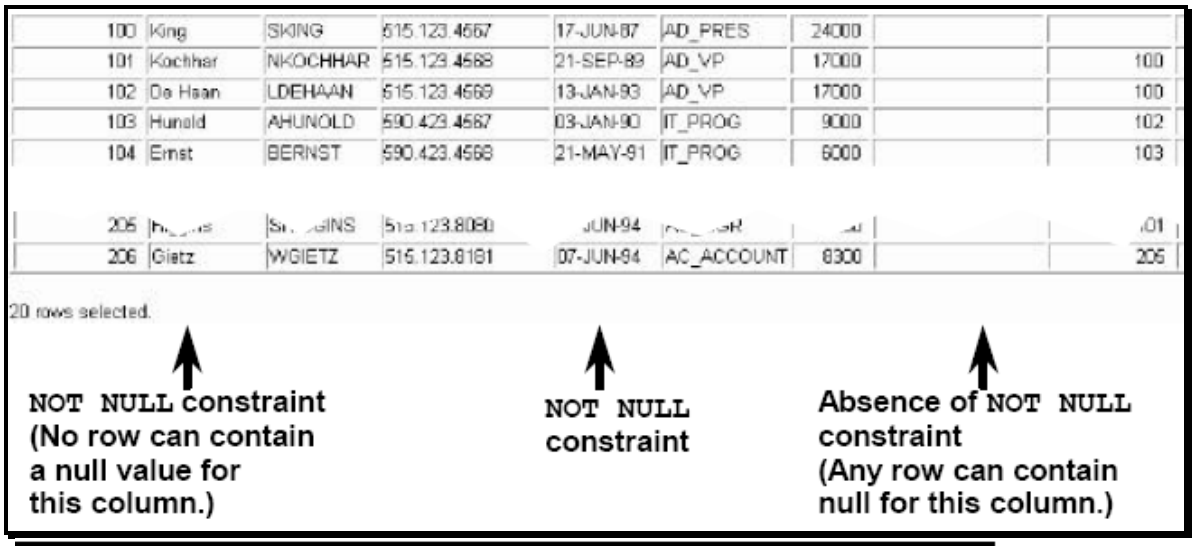
مثال:

```
CREATE TABLE  emp
(
    emp_id          NUMBER (6) ,
    emp_name        VARCHAR2 (20) ,
    work_id          VARCHAR2 (10)   NOT NULL ,
    CONSTRAINT  emp_id_pk    PRIMARY KEY  (emp_id) );
```

لاحظ كيف تم تطبيق القيد NOT NULL على مستوى العمود work_id، و القيد PRIMARY KEY على مستوى الجدول.

القيد NOT NULL :

يُفرض هذا القيد على الأعمدة التي نريدها ألا تتضمن قيم فارغة، ويتم فرضه على مستوى العمود فقط، ولا يمكن استخدامه على مستوى الجدول.



100	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000		
101	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000		100
102	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000		100
103	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000		102
104	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000		103
205	[illegible]	[illegible]	515.123.8080	JUN-94	[illegible]	[illegible]		101
206	Gietz	WGIETZ	515.123.8181	07-JUN-94	AC_ACCOUNT	8300		205

20 rows selected.

NOT NULL constraint (No row can contain a null value for this column.)

NOT NULL constraint

Absence of NOT NULL constraint (Any row can contain null for this column.)

```
CREATE TABLE employees(
    employee_id    NUMBER(6),
    last_name      VARCHAR2(25) NOT NULL,
    salary         NUMBER(8,2),
    commission_pct NUMBER(2,2),
    hire_date      DATE
                   CONSTRAINT emp_hire_date_nn
                   NOT NULL,
...
```

System named

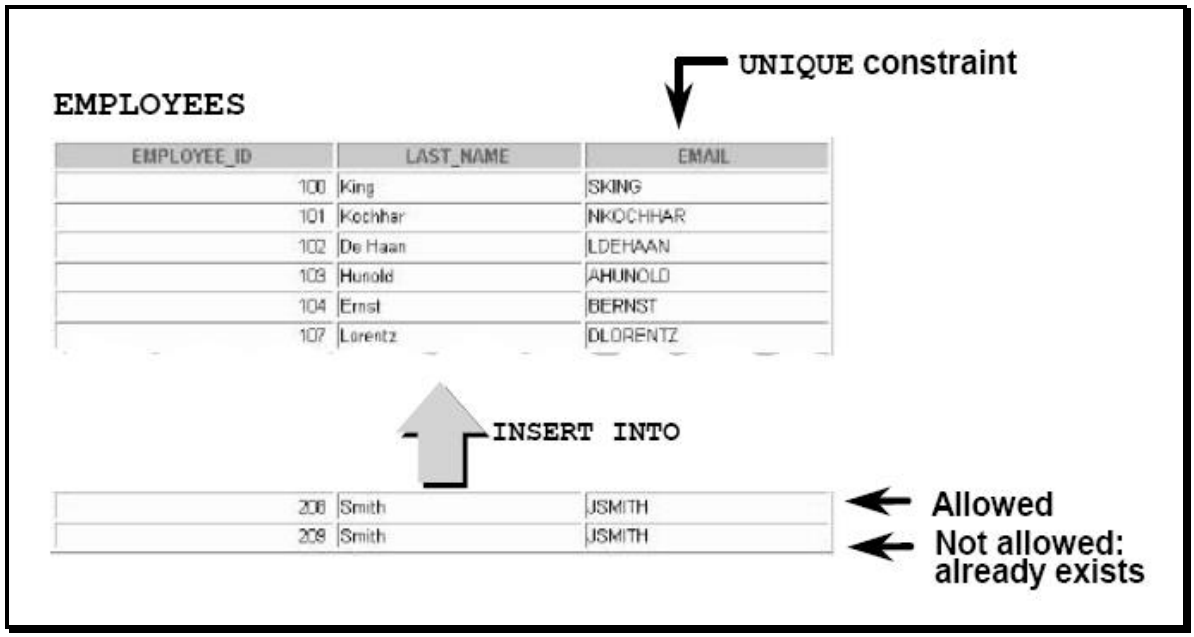
User named

لاحظ أن القيد الأول لم يتم تسميته من قبل المستخدم، لذا فإن نظام إدارة قواعد البيانات يتكفل بهذه المهمة، ويولد اسماً داخلياً لهذا القيد، أما القيد الثاني فقد تمت تسمية القيد صراحة من قبل المستخدم، وهي الطريقة الأفضل.

القيد UNIQUE:

يُفرض على الأعمدة التي لانريدها أن تتضمن قيم مكررة، ويجب الأخذ بعين الاعتبار أن الأعمدة التي يفرض عليها هذا القيد يمكن أن تتضمن قيم فارغة NULL، إلا في حال فرض قيد NOT NULL عليها أيضاً.

يمكن فرض هذا القيد على مستوى العمود، وكذلك على مستوى الجدول.

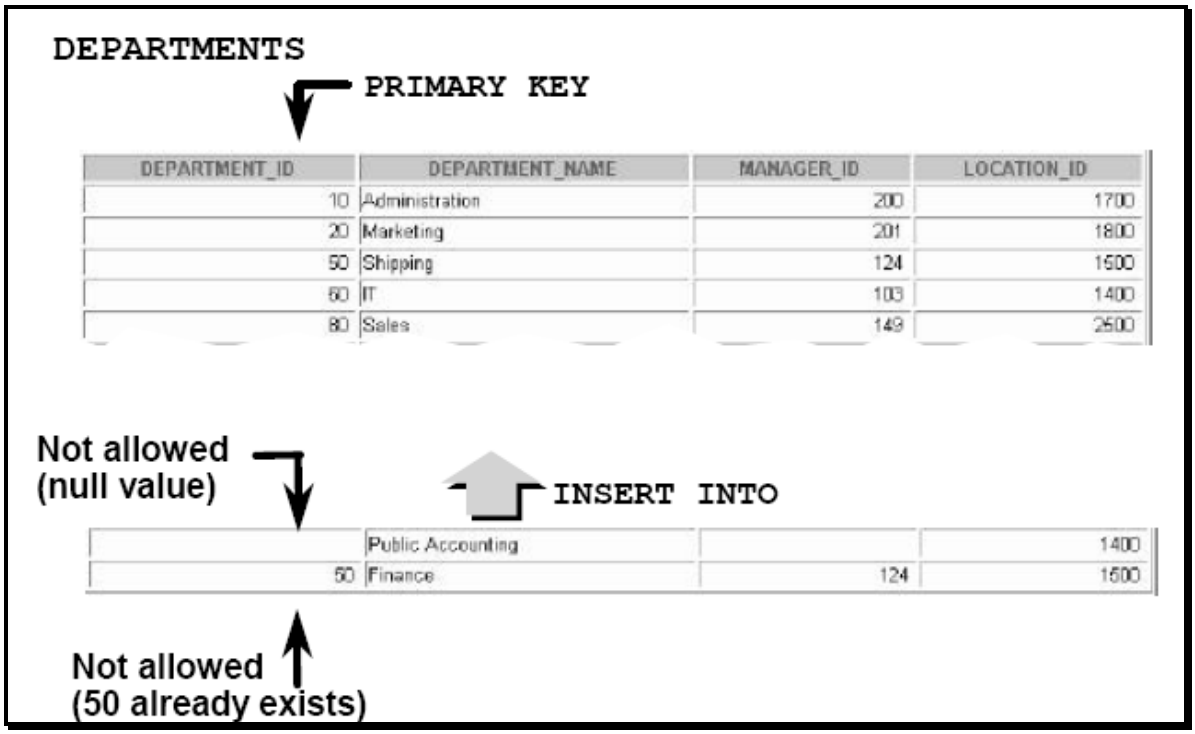


مثال:

```
CREATE TABLE employees(
    employee_id      NUMBER(6),
    last_name        VARCHAR2(25) NOT NULL,
    email            VARCHAR2(25),
    salary           NUMBER(8,2),
    commission_pct   NUMBER(2,2),
    hire_date        DATE NOT NULL,
...
    CONSTRAINT emp_email_uk UNIQUE(email));
```

القيد PRIMARY KEY:

يُسمح باستخدام هذا القيد مرة واحدة لكل جدول، ويُفرض على العمود أو الأعمدة التي نريدها أن تتضمن قيم غير مكررة وليست فارغة، ويعرّف على مستوى العمود أو على مستوى الجدول، حيث يفرض على مستوى الجدول عند الحاجة لفرضه على عدة أعمدة.



مثال:

```
CREATE TABLE   departments(
    department_id         NUMBER(4),
    department_name       VARCHAR2(30)
      CONSTRAINT dept_name_nn NOT NULL,
    manager_id            NUMBER(6),
    location_id           NUMBER(4),
      CONSTRAINT dept id pk PRIMARY KEY(department id));
```

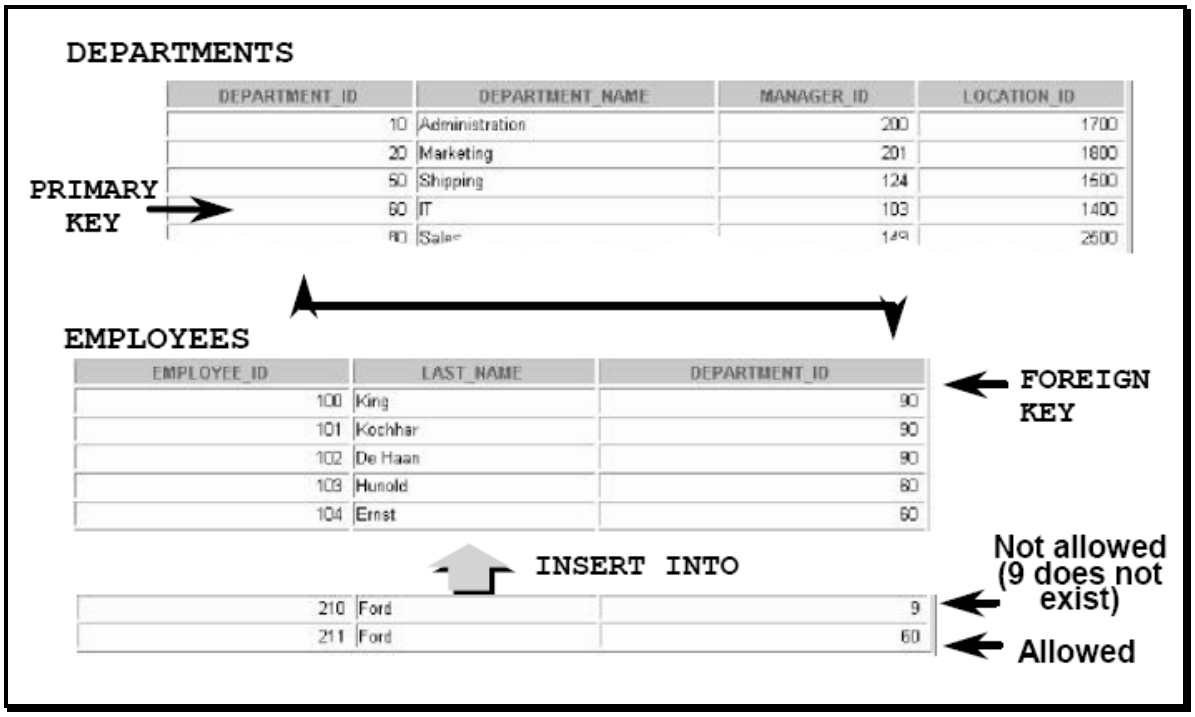
القيد FOREIGN KEY:

يمكن فرضه على مستوى العمود أو على مستوى الجدول.

يستخدم مع هذا القيد الكلمة المحجوزة REFERNCES التي تحدد الجدول الأب والعمود الموافق من ذلك الجدول.

يملك هذا القيد الخاصتين التاليتين:

- **ON DELETE CASCADE: Deletes the dependent rows in the child table when a row in the parent table is deleted**
- **ON DELETE SET NULL: Converts dependent foreign key values to null**



مثال:

```
CREATE TABLE employees(
    employee_id      NUMBER(6),
    last_name        VARCHAR2(25) NOT NULL,
    email            VARCHAR2(25),
    salary           NUMBER(8,2),
    commission_pct   NUMBER(2,2),
    hire_date        DATE NOT NULL,
...
    department_id    NUMBER(4),
    CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)
      REFERENCES departments(department_id),
    CONSTRAINT emp_email_uk UNIQUE(email));
```

ويمكن تعريف القيد السابق على مستوى العمود بالشكل التالي:

```
CREATE TABLE employees
(...
department_id NUMBER(4) CONSTRAINT emp_deptid_fk
    REFERENCES departments(department_id),
...
)
```

القيد CHECK:

يحدد شرط يجب على كل سطر في الجدول أن يحققه، لا يمكن استخدام التوابع مثل SYSDATE في هذا القيد، وكذلك لا يمكن استخدام استعلامات تشير إلى قيم أخرى في صفوف أخرى.

مثال:

```
CREATE TABLE employees
     (...
      salary NUMBER(8,2) CONSTRAINT emp_salary_min
                         CHECK (salary > 0),
     ...
```

يمكن استخدام تعليمة ALTER TABLE من أجل:

- إضافة أو حذف قيد، لكن لا يمكن تعديل هذا القيد.
- تفعيل أو عدم تفعيل القيد.
- إضافة القيد NOT NULL عن طريق استخدام العبارة MODIFY، وهذا فقط في حال كان الجدول لا يحوي بيانات بعد، أو العمود يتضمن قيم من أجل كل سطر في الجدول.

الشكل العام:

```
ALTER TABLE table
ADD [CONSTRAINT constraint] type (column);
```

مثال 1:

```
ALTER TABLE      employees
ADD CONSTRAINT   emp_manager_fk
  FOREIGN KEY(manager_id)
  REFERENCES employees(employee id);
```

مثال 2:

```
ALTER TABLE      employees
DROP CONSTRAINT  emp_manager_fk;
```