

אילוצים (Constraints)

- מגדירים כללים שיחולו על הנתונים המאוחסנים בעמודות ואוכפים אותם כדי להגן על הנתונים. בסיס נתונים לא יאפשר לאחסן בו נתונים המפרים את כללי האילוץ.

סוגי אילוצים:

NOT NULL .1	DEFAULT .4
PRIMARY KEY .2	CHECK .5
UNIQUE .3	FOREIGN KEY .6

1. NOT NULL

- עמודה לא יכולה להישאר ריקה

```
CREATE TABLE Tab  
(A NUMBER(3),  
B VARCHAR2(10) NOT NULL,  
.....);
```

- א:ו:

```
CREATE TABLE Tab  
(A NUMBER(3),  
B VARCHAR2(10) ,  
.....  
B CONSTRAINT tab_1 NOT NULL);
```

- אחרי יצירת הטבלה:

```
ALTER TABLE Tab MODIFY(B VARCHAR2(10) NOT NULL);
```

2. PRIMARY KEY

- מפתח ראשי – אוסף תכונות (עמודות) אשר בצורה חד-משמעית מזהה כל רשומה בטבלה.

- מפתח ראשי מכיל תכונות:

- ערך ייחודי (לא חוזר על עצמו)

- לא יכול להכיל ערך NULL(ריק)

- הגדרת מפתח ראשי בזמן יצירת הטבלה:

```
CREATE TABLE Students(StudID NUMBER(6) PRIMARY KEY,  
Name VARCHAR2(15),
```

.....
);

אם המפתח הראשי מורכב מכמה עמודות אז יש ליצור אותו בצורה הבאה:

```
CREATE TABLE CoursForStud (StudID NUMBER(6),
CoursID VARCHAR2(15),
Mark NUMBER(2),
.....,
PRIMARY KEY(StudID, CoursID)
);
```

```
CREATE TABLE CoursForStud (StudID NUMBER(6),
CoursID VARCHAR2(15),
Mark NUMBER(2),
.....
CONSTRAINT pk_stud PRIMARY KEY(StudID, CoursID)
);
```

- pk_stud שם לאילוף של מפתח ראשי.

הגדרת מפתח ראשי אחרי יצירת הטבלה:

```
ALTER TABLE Students ADD PRIMARY KEY(StudID);
ALTER TABLE Students ADD CONSTRAINT pk_Stud PRIMARY
KEY(StudID);
```

UNIQUE .3

-ערך ייחודי (לא חוזר על עצמו) אבל יכול להישאר ריק .
בכל טבלה ניתן להגדיר פעם אחת אילוף מסוג PRIMARY KEY אבל מסוג UNIQUE ניתן להגדיר כמה שדות .

```
CREATE TABLE Tab
(A NUMBER(3),
B VARCHAR2(10) UNIQUE,
.....
E DATE UNIQUE);
```

DEFAULT

הגדרת ערכים כברירת המחדל- במקרה שלא נכניס שום ערך לשדה שהוגדר עם DEFAULT השדה יציב את הערך שהוגדר כ- DEFAULT וכך ערך השדה לא יהיה NULL.
CREATE TABLE Students
(A NUMBER(3),
Gender CHAR(1) **DEFAULT** '?',
City VARCHAR2(15) **DEFAULT** 'BEER_SHEVA',
.....);

או אחרי יצירת הטבלה:

ALTER TABLE Students ADD mobil VARCHAR2(10) DEFAULT 'UNLISTED';

CHECK

הגדרת ערכים חוקיים לעמודה. כל ניסיון לשנות או להכניס ערכים שמפרים את הכלל הזה יכשלו ותתקבל הודעת שגיאה.
הגדרת אילוץ בזמן בניית הטבלה:

.1

CREATE TABLE temp
(.....,
D DATE CHECK (d>to_date('01-JAN-2003','DD-MON-YYY')),
E VARCHAR2(10) CONSTRAINT check_e CHECK(e='Z'));

.2

CREATE TABLE retire_women
(gender char;
AGE NUMBER(3) ,
CONSTRAINT ret_chk **CHECK** (gender ='f' and age>60);

ניסיון להכניס רשומות שסותרות את האילוץ:

INSERT INTO temp VALUES(.....,'01-JAN-2001','Z');
// error - '01-JAN-2001'
INSERT INTO temp VALUES (.....,'02-FEB-2003','P');
//error – P (!=Z)

הגדרת אילוץ לאחר בניית הטבלה:

1. ALTER TABLE dugma ADD CHECK (CustomerID>0);
2. ALTER TABLE prices ADD **CONSTRAINT** pri_sd_chk
CHECK (StartDate<=EndDate);

FOREIGN KEY

עמודה שמוגדרת כמפתח זר יכולה לקבל רק ערכים שמוגדרים ב- references למשל, ניתן להכניס לעמודה deptno מס' מחלקה לעובד רק אחד מהערכים מ- deptno מטבלת מחלקות (dept). כמובן, הטבלה dept חייבת להיווצר לפני emp.

```
CREATE TABLE Emp
(emp_id NUMBER(3) PRIMARY KEY,
.....
dept_no NUMBER(3),
CONSTRAINT fk_emp FOREIGN KEY (dept_no)
REFERENCES dept(deptno));
```

או אחרי יצירת הטבלה:

```
ALTER TABLE emp ADD CONSTRAINT fk_dep FOREIGN KEY (deptno)
REFERENCES dept (deptno);
```

- מפתח זר מתייחס לשדה/שדות בטבלה אחרת. השדות האלו חייבים להיות מפתח ראשי בטבלה שבה נמצאים.
- לא ניתן להוסיף ערכים למפתח הזר שלא מופיעים בשדה/ות בטבלה אליה מתייחס המפתח.

לקבלת מידע על primary key ו-foreign key כהגבלות לטבלה :

```
SELECT owner, constraint_name, table_name, constraint_type
FROM user_constraints WHERE table_name='EMP';
```

לקבלת מידע על הגבלות על שדות :

```
SELECT * FROM user_cons_columns WHERE table_name='EMP';
```

שם הטבלה חייב להיות באותיות גדולות.

Referential integrity

במקרים הבאים מתקבלת run-time error:

1. מכניסים עובד חדש עם מספר מחלקה לא קיים בטבלת מחלקות
2. מנסים לעדכן לעובד ערך של שדה מס' מחלקה לערך חדש אשר לא מופיעה ברשימת מחלקות.
3. מנסים למחוק מטבלת מחלקות נתונים על מחלקה (למשל 10) אשר קיימים במחלקה עובדים (ז"א בטבלת עובדים יש רשומות שבשדה deptno מופיעה ערך 10)
4. מנסים לעדכן מספר מחלקה בטבלת מחלקות (למשל 10 מעדכנים ל-50) אשר קיימים במחלקה עובדים (ז"א בטבלת עובדים יש רשומות שבשדה deptno מופיע ערך 10)

עמודה שמוגדרת כמפתח זר יכולה לקבל רק ערכים שמוגדרים ב- references למשל, ניתן להכניס לעמודה deptno (מס' מחלקה לעובד) רק אחד מהערכים מ- deptno מטבלת מחלקות (dept).

ברגע שיוצרים מפתח זר אשר מתייחס לטבלה אחרת לא ניתן יהיה למחוק את הטבלה (אחרת)
או חלק ממנה. למשל,

```
DELETE FROM dept;
```

Error: integrity constraint (SCOTT.FK_DEPNO) violated – child record found.

```
DROP TABLE dept;
```

Error: unique/primary key in table referenced by foreign key;

אין בעיה בפקודה הבאה:

```
DROP TABLE emp;
```

מחיקת טבלאות שיש בהן קשר ע"י מפתח זר צריך לבצע בסדר הבא:
קודם למחוק טבלה שמוגדר בה מפתח זר (מצד many) ורק אחר כך טבלה ראשית (מצד one)
למשל, קודם יש למחוק emp ורק אחר כך dept

```
DROP TABLE emp;
```

```
DROP TABLE dept;
```

אופציה ON DELETE CASCADE for FOREIGN KEY

```
CREATE TABLE Emp
```

```
(emp_id NUMBER (3) PRIMARY KEY,
```

```
.....
```

```
dept_no NUMBER (3),
```

```
CONSTRAINT fk_emp FOREIGN KEY (dept_no) REFERENCES dept (dno))
```

```
ON DELETE CASCADE);
```

אם מוחקים שורה ב-dept (בטבלה הראשית) שורות תואמות נמחקים באופן אוטומטי
מ-emp (בטבלה המשנית)

אופציות הבאות לא עובדות ב-Oracle:

עדכון מדורג - ON UPDATE CASCADE

השמת ערך NULL לשדה רלוונטי ברשימות הרלוונטיות - אופציה :

ON DELETE SET NULL

מחיקת הגבלות :

```
ALTER TABLE Students DROP CONSTRAINT pk_stud;
```

לבדוק את השינוי:

```
SELECT * FROM user_constraints WHERE table_name='STUDENTS';
```

Output:

OWNER	CONSTRAINT_NAME	C	TABLE_NAME
STUDENT1	pk_stud	p	STUDENTS

אופציה מחיקה מדורגת – CASCADE:

נראה קודם את האילוצים של הטבלאות EMP ו DEPT

```
SELECT TABLE_NAME , CONSTRAINT_NAME ,R_CONSTRAINT_NAME
FROM user_constraints
WHERE table_name IN ('EMP','DEPT');
```

TABLE_NAME	CONSTRAINT_NAME	R_CONSTRAINT_NAME
DEPT	SYS_C004791	-
DEPT	DEPT_PRIMARY_KEY	-
EMP	SYS_C004793	-
EMP	SYS_C004794	-
EMP	EMP_EMPNO_PK	-
EMP	EMP_MGR_FK	EMP_EMPNO_PK
EMP	EMP_DEPTNO_FK	DEPT_PRIMARY_KEY

כעת נמחק את האילוך DEPT_PRIMARY_KEY שהוא המפתח הראשי של הטבלה DEPT, ומכיוון שבטבלה EMP יש מפתח חיצוני אליו, אז גם הוא ימחק.

```
ALTER TABLE dept DROP CONSTRAINT DEPT_PRIMARY_KEY
CASCADE;
```

עכשיו נתבונן בשינויים.

```
SELECT TABLE_NAME , CONSTRAINT_NAME ,R_CONSTRAINT_NAME
FROM user_constraints
WHERE table_name IN ('EMP','DEPT');
```

TABLE_NAME	CONSTRAINT_NAME	R_CONSTRAINT_NAME
DEPT	SYS_C004791	-
EMP	SYS_C004793	-
EMP	SYS_C004794	-
EMP	EMP_EMPNO_PK	-
EMP	EMP_MGR_FK	EMP_EMPNO_PK

עבודת כיתה:

עבודה 1:

נתונה הסכמה הבאה:

Author(Auld,AuthName,born,died,)
Books(ISBN,title,pages,pubid,price)
Publ(pubid,pubname,address)
AuBook(Auid,ISBN)

צרו את הטבלאות עם הגדרות נכונות של מפתחות ראשיים וזרים.

בנו את האילוצים הבאים:

לטבלה Author:

* למחבר חייב להיות שם .

* תאריך לידה חייב להיות לפני תאריך הפטירה.

לטבלה Books:

* לא יתכנו שני ספרים בעלי אותו שם .

* מספר דפים חייב להיות חיובי .

* מחיר חייב להיות חיובי .

לטבלה Publ :

*(pubname) שם המוציא לאור לא יכול לחזור על עצמו .

* Address : אם לא ניתנת כתובת יש להכניס 'unknown'

alter session set nls_date_format = 'dd/mm/yyyy';

עבודה 2:

הוסיפו מפתחות זרים לטבלאות שבניתם.

Author(Auld,AuthName,born,died,)
Books(ISBN,title,pages,pubid,price)
Publ(pubid,pubname,address)
AuBook(Auid,ISBN)

1. הגדירו את ה- TYPE של PRODUCT כאחד מהערכים הבאים בלבד:
PC,LAPTOP,PRINTER

2. PC עם מהירות קטנה מ- 1200 לא יעלה יותר מ- \$1500.

3. LAPTOP עם מסך בגודל פחות מ-15 חייב להיות עם 20 גיגה או לעלות פחות מ-
\$2000.