

צירוף של טבלאות

1. מכפלה קרטזית (Cartesian product)

צירוף טבלאות ללא הגבלה
למשל, WHERE

SELECT * FROM T1;

row	rem
R1	T1
R2	T1
R3	T1
R4	T1

SELECT * FROM T2;

row	rem
R1	T2
R2	T2
R3	T2
R4	T2

SELECT * FROM T1,T2;

<u>T1.row</u>	<u>T1.rem</u>	<u>T2.Row</u>	<u>T2.rem</u>
R1	T1	R1	T2
R1	T1	R2	T2
R1	T1	R3	T2
R2	T1	R1	T2
R2	T1	R2	T2
R2	T1	R3	T2
R3	T1	R1	T2
R3	T1	R2	T2
R3	T1	R3	T2

צירוף מסוג הזה מחבר את כל השרות מכל הטבלאות המוכרזות בהוראה FROM. מה שאנו מקבלים זו מכפלה קרטזית של הטבלאות.

2. צירוף רגיל על בסיס שיוון (equal join)

צירוף טבלאות עם הגבלה WHERE בתאום של שדות קשר למשל,

```
SELECT ename,sal, emp.dname dname
FROM emp,dept
WHERE emp.deptno=dept.deptno;
```

כאשר יש שדות עם שמות זהים צריך לזהות ע"י שם הטבלה.

ENAME	SAL	DEPTNO
KING	5000	10
BLAKE	2850	30
CLARK	2450	10
JONES	2975	20
MARTIN	1250	30
ALLEN	1600	30
TURNER	1500	30
JAMES	950	30
WARD	1250	30
FORD	3000	20

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

חלק מהטבלה:

ENAME	SAL	EMP.DEPTNO	DEPT.DEPTNO	DNAME
KING	5000	10	10	ACCOUNTING
BLAKE	2850	30	10	ACCOUNTING
CLARK	2450	10	10	ACCOUNTING
JONES	2975	20	10	ACCOUNTING
MARTIN	1250	30	10	ACCOUNTING
ALLEN	1600	30	10	ACCOUNTING
TURNER	1500	30	10	ACCOUNTING
JAMES	950	30	10	ACCOUNTING
WARD	1250	30	10	ACCOUNTING
FORD	3000	20	10	ACCOUNTING

ממנה נקבל:

ENAME	SAL	DNAME
KING	5000	ACCOUNTING
BLAKE	2850	SALES
CLARK	2450	ACCOUNTING
JONES	2975	RESEARCH
MARTIN	1250	SALES
ALLEN	1600	SALES
TURNER	1500	SALES
JAMES	950	SALES
WARD	1250	SALES
FORD	3000	RESEARCH

זה צירוף שיווני כי מטרתו להתאים בין ערכי העמודות בטבלה אחת לבין ערכים המתקיימים בטבלה אחרת.

ניתן לכנות את הטבלאות בשם ולהשתמש בו לאורך כל השאלתה:

```
SELECT ename, sal, e.dname  
FROM emp e, dept d  
WHERE e.deptno=d.deptno;
```

ניתן גם להגדיר מספר קריטריונים :

```
SELECT ename,sal,name  
FROM emp,dept  
WHERE emp.deptno=dept.deptno  
AND sal>2000;
```

במקרה של חיבור יותר מ-2 טבלאות צריך להקפיד על קישור נכון בין כל הטבלאות:

```
SELECT c.custid, c. name ,( i. qty * i.actualprice) total  
FROM ord o, item i ,customer c  
WHERE o. ordID = i. ordID AND  
o.custid = c.custid;
```

צירוף חיצוני (outer join)

"הצג לי הכל , גם אם חסר דבר מה"

למשל, אם לא כל הלקוחות בצעו הזמנה – נקבל רשימת של כל הלקוחות ללא קשר לביצוע הזמנות.

את הסימן (+) רושמים לצד העמודה שבה עשויים להיות נתונים חסרים מול העמודה איתה מתבצע הצירוף.

לקוחות והזמנות שעשו. יופיעו גם לקוחות שלא ביצעו שום הזמנה.

```
SELECT c.name,o.ordid  
FROM customer c,ord o  
WHERE c.custid=o.custid(+);
```

נכניס את הרשומה הבאה שהיא לקוח חדש וברור שהוא עדיין לא ביצע הזמנות:

```
INSERT INTO CUSTOMER (ZIP, STATE, REPID, PHONE, NAME, CUSTID,  
CREDITLIMIT, CITY, AREA, ADDRESS, COMMENTS)  
VALUES ('96711', 'CA', '7844', '598-6609', 'sce', '1000', '5000', 'BELMONT', '415',  
'345 VIEWRIDGE',
```

'Very friendly people to work with -- sales rep likes to be called Mike.');

בביצוע השאילתה הבאה נקבל שדה ריק ליד שם הלקוח כלומר אין לו הזמנה משוייכת.

```
SELECT c.name,o.ordid
FROM customer c,ord o
WHERE c.custid=o.custid(+);
```

NAME	ORDID
TKB SPORT SHOP	610
VOLLYRITE	611
SHAPE UP	612
JOCKSPORTS	606
JOCKSPORTS	609
EVERY MOUNTAIN	607
EVERY MOUNTAIN	608
VOLLYRITE	603
JOCKSPORTS	620
NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER	613
VOLLYRITE	614
JUST TENNIS	616
EVERY MOUNTAIN	619
K + T SPORTS	617
WOMENS SPORTS	615
VOLLYRITE	618
JOCKSPORTS	621
sce	-

צירוף עצמי (self join)

אנו רוצים למצוא לכל עובד את המנהל שלו:

```
SELECT a.ename empl ,b.ename Mangr
FROM emp a,emp b
WHERE b. empno =a. mgr
ORDER BY Mangr;
```

ORDER BY Mangr;
אנו מצרפים טבלה לעצמה:

EMPNO	ENAME	MGR
7839	KING	-
7698	BLAKE	7839
7782	CLARK	7839
7566	JONES	7839
7654	MARTIN	7698

EMPNO	ENAME	MGR
7839	KING	-
7698	BLAKE	7839
7782	CLARK	7839
7566	JONES	7839
7654	MARTIN	7698

A.EMPNO	A.ENAME	A.MGR	B.EMPNO	B.ENAME	B.MGR
7566	JONES	7839	7839	KING	-
7782	CLARK	7839	7839	KING	-
7698	BLAKE	7839	7839	KING	-
7521	WARD	7698	7698	BLAKE	7839
7900	JAMES	7698	7698	BLAKE	7839

אנו רוצים למצוא לכל עובד את המנהל שלו:

```
SELECT a.ename empl ,b.ename Mangr
FROM emp a,emp b
WHERE b. empno =a. mgr
ORDER BY Mangr;
```

Output:

EMPL	MANGR
JONES	KING
CLARK	KING
BLAKE	KING
WARD	BLAKE
JAMES	BLAKE

3. צירוף טבלאות על סמך אי-שוויון

להציג את כל העובדים אשר מקבלים משכורת גבוהה יותר מהמשכורת של עובד מס' 7782:

```
SELECT t1.empno,t1.ename
FROM emp t1,emp t2
WHERE t1.sal>t2.sal
AND t2.empno=105;
```

6.איחוד (union)

באיחוד נתונים נקבל מאגר רשומות שיכיל את כל איברי הטבלה הראשונה והשנייה – ללא חזרות.

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S1	Smith	20	London
S4	Clark	20	London

A

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S1	Smith	20	London
S2	Jones	10	Paris

B

A UNION B :

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S1	Smith	20	London
S4	Clark	20	London
S2	Jones	10	Paris

למשל,

```
SELECT S#, SName,status,City
FROM A
UNION
SELECT S#, SName,status,City
FROM B;
```

הערה: משפטי SELECT שמטפלים באיחוד, חיתוך ומינוס חייבים להכיל מספר עמודות, טיפוסים וסידור זהים.

:UNION ALL

```
SELECT S#, SName,status,City
FROM A
UNION ALL
SELECT S#, SName,status,City
FROM B
```

A UNION ALL B :

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S1	Smith	20	London
S4	Clark	20	London
S1	Smith	20	London
S2	Jones	10	Paris

האופציות הבאות לא עובדות ב-ORACLE : MINUS ALL , INTERSECT ALL

7. חיתוך (intersect)

מחזיר את כל הרשומות הזהות. אלו שנמצאות בתוצאת השאילתה הראשונה וגם בתוצאת השאילתה השנייה.

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S1	Smith	20	London
S4	Clark	20	London

A

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S1	Smith	20	London
S2	Jones	10	Paris

B

A INTERSECT B:

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S1	Smith	20	London

```
SELECT S#, SName,status,City
FROM A
INTERSECT
SELECT S#, SName, status ,City
FROM B;
```

8. הפרש (minus)

מחזיר רשומות שנמצאות בתוצאת השאילתה ראשונה ולא נמצאות בתוצאת השאילתה השנייה.

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S1	Smith	20	London
S4	Clark	20	London

A

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S1	Smith	20	London
S2	Jones	10	Paris

B

A MINUS B:

A:

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S4	Clark	20	London

```
SELECT S#, SName,status,City
FROM A
MINUS
SELECT S#, SName,status,City
FROM B;
```

B MINUS A:

<u>S#</u>	<u>SNAME</u>	<u>STATUS</u>	<u>CITY</u>
S2	Jones	10	Paris

```
SELECT S#, SName,status,City
FROM B
MINUS
```

SELECT S#, SName,status, City
FROM A;

תרגיל במעבדה מס' 3 בבסיסי נתונים

יש לבנות את השאילתות הבאות:

1. להציג את כל המחשבים אישיים (מודל, מהירות, דיסק קשיח) שמחירם מעל \$1600 וגם גודל ה-hd שלהם הוא 10 או 20.
 2. להציג את כל המחשבים אישיים (מודל, מהירות, דיסק קשיח) שמחירם מעל \$1600 את עמודת המהירות יש להציג כ-megahertz ואת עמודת הדיסק הקשיח כ-gigabytes.
 3. להציג את כל יצרני המחשבים אישיים.
 4. להציג את כל המחשבים הניידים (מודל, זיכרון, מסך) אשר מחירם מעל \$2000 והמסך שלהם לפחות בגודל 14" או גודל הזיכרון לפחות 64 MB.
 5. למחשבים ניידים עם דיסק קשיח של לפחות 1 GB יש להציג את שמות היצרנים ואת מהירות המחשב.
 6. להציג את כל המודלים של המחשביים האישיים אשר מהירים יותר ממהירותו של מחשב מודל 1001.
 7. יש להציג את מספר המודל והמחיר של כל המוצרים (מכל סוג שהוא) שייצר אותם יצרן בשם 'A'. יש להשתמש בפיתרון באופרטור UNION.
 8. יש להציג את כל המחשבים האישיים וגם המחשבי הניידים (מודל, מהירות, מחיר) אשר מחירם עולה על \$2000 וגם מהירותם גבוהה מ-100. יש להשתמש בפיתרון באופרטור UNION.
 9. יש להציג את כל היצרנים שמייצרים מחשבים אישיים וגם מחשבים ניידים אשר מחירם עולה על \$2000 וגם מהירותם גבוהה מ-100. יש להשתמש בפיתרון באופרטור INTERSECT.
 10. צרו טבלת העתק לטבלה dept בשם copy_dept. יש להכניס לטבלה copy_dept רשומה חדשה:
- 50 CHEKING BOSTON
- יש להציג בשאילתה אחת את שם המחלקה, שם העובד והמשכורת, לעובדים שהמחלקה שלהם נמצאת ב-BOSTON or CHICAGO. יש לקבל נתונים של כל המחלקות הקיימות אשר עומדות בקריטריון הנ"ל גם אם למחלקה לא מוגדר עדיין אף עובד.
11. יש למצוא את היצרנים שמייצרים מחשבים ניידים אבל לא מייצרים מחשבים אישיים.

12. יש למצוא את גודלי הדיסק הקשיח שמופיעים בשני מחשבים אישיים או יותר.

13. יש למצוא את זוגות המודלים של מחשבים אישיים שיש להם אותה מהירות ואותו

RAM. כל זוג חייב להופיעה פעם אחד בלבד!

בהצלחה !!!