

(ii) Left Join:

Give me all the records in the table on the left regardless of it is matched or not in the right table

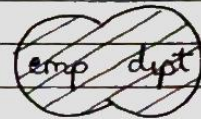
```
select *  
from emp Left Outer Join dept  
on emp.deptno = dept.deptno
```

IS SAME AS

```
select *  
from emp, dept  
where emp.deptno = dept.deptno (+)
```

NOT RECOMMENDED
(Oracle oriented only)

(iii) Full Outer Join



It will join where it can or otherwise will put NULL in left as well as in Right unmatched records.

```
① select *  
from emp Full Outer Join dept  
on where emp.deptno = dept.deptno
```

② Sel Using Subqueries:

```
select *  
from (select * from emp) e  
Full Outer Join  
(select * from dept) d
```

on e.deptno = d.deptno



NOTE: EXISTS is not an efficient way to write SQL queries.



EXISTS:

SYNTAX:

WHERE EXISTS (select * from ...)

Ex1: { select *
from emp }

where EXISTS (select * from emp
where job = 'Artist')

(If the Exists statement gets true then the { } statement gets executed)

NOT EXISTS:

Ex2: select *
from emp

where NOT EXISTS
(select * from dept
where loc = 'Mumbai')



PRIMARY KEY :

The Primary Key constraints uniquely identifies each record in a table.

Primary Keys must contain UNIQUE values, and ~~or~~ cannot contain NULL values.

A table can have only ONE primary key.

Primary key can consist of single or multiple columns (fields).

NOTE: Apex Grade application is AUTO commit.



Creating table store :

```
CREATE TABLE stores
```

```
(  
    store_id number not null,  
    city varchar(50)  
);
```

* INSERT

Inserting individual records :

```
INSERT INTO stores (store_id, city)  
VALUES (1, 'San Francisco');
```

```
Insert into stores (store_id, city)  
values (2, 'New York City');
```

```
Insert into stores (store_id, city)  
values (3, 'Chicago');
```

Inserting Multiple records
in One Insert statement:

Insert all

into stores (store-id, city)
values (4, 'Philadelphia')

into stores (stores-id, city)
values (5, 'Boston')

into ^{stores} values (store-id, city)
values (6, 'Seattle')

SELECT * From Dual;

NOTE: (It requires some kind of
select statement at the end.)

Creating table (products)
with a Primary Key constraint:

Create Table products

(
product-id number not null,
name varchar2(50),
product-cost number(5,2),
product-retail number(5,2),
product-type varchar2(10),
store-id number not null,

CONSTRAINT product-pk
PRIMARY KEY (product-id)
)

NOTE:

If we are inserting a record for every column of a table THEN we do not need to explicitly specify the column names of the table.

Insert into products
(product-id, name, product-cost, product-retail, product-type, store-id)

Values (1001, 'Colgate Toothpaste', 2.25, 5.47, 'Hygiene', 2)

IS SAME AS

Insert into products Values
(1001, 'Colgate Toothpaste', 2.25, 5.47, 'Hygiene', 2)

ALTER TABLE STATEMENT:

The Alter Table statement is used to add, delete or modify columns in an existing table.

The Alter Table statement is also used to add and drop various constraints on an existing table.

(1) ALTER TABLE - ADD Column

Syntax:

Alter Table products
ADD ^{column} expiry-date date;

(2) ALTER TABLE - DROP Column

Syntax:

Alter table products
DROP ^{column} ~~column~~ expiry-date;

(3) ALTER TABLE - MODIFY Column

To change the data type
of a column in a table.
(ADD & DROP constraints too)

Syntax: (Check: Alter table - drop column)

```
Alter table products  
MODIFY name varchar2(50)  
not null;
```

Modifying multiple columns:

```
Alter table products
```

```
Modify ( product-cost number(5,2)  
not null,  
product-retail number(5,2)  
not null );
```

RENAME a column

Alter table products

Rename column name to
product-name;

Creating table using SELECT statement

Create table employees AS

SELECT empno, ename, job,
hiredate, sal, comm
FROM emp;

(employees is borrowing
structure as well as
data from emp table)

UPDATE:

The UPDATE statement is used
to modify the existing records
in a table.

Syntax:

UPDATE table-name
SET column1 = value1, column2 = value2,..
WHERE condition,

NOTE: If we omit the WHERE
clause, all the records
in the table will be
updated!

UPDATE employees
SET store-id = 3
WHERE ename in ('KING',
 'BLAKE', 'CLARK')

* DELETE Statement:

The DELETE statement is used to delete existing records in a table.

Syntax:

DELETE FROM Table-name
WHERE condition;

NOTE:

If you omit the WHERE clause, all records in the table will be deleted!

Ex1: Delete from dept
where deptno = 40

Ex2: Delete from emp
where ename IN ('Blake',
'King', 'Jones')

Ex3: Dropping Foreign Key constraint:

Alter Table emp

Drop Constraint SYS-C0044995584

* DROP TABLE Statement:

The DROP TABLE statement is used to drop an existing table in a database.

Syntax:

DROP TABLE table-name

Ex1: Drop table dept

NOTE:

The DROP TABLE statement completely wipes out the DATA as well as the STRUCTURE too.

* TRUNCATE TABLE

The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

Syntax:

TRUNCATE TABLE table-name;

Ex1: Truncate table emp;

→ No data found (for (select * from emp))

SQL INDEX

Indexes are used to retrieve data from the database very fast. The users cannot see the indexes, they are just used to speed up searches / queries (eg: like a index in a book)

NOTE:

Only create indexes on columns that will be frequently searched against.

(1) CREATE INDEX :

Syntax:

```
Create [Unique] Index index-name  
On table-name (column1, column2;...)  
[Compute statistics];
```

Ex1: Create Index emp-name-idx
On employees (ename)

```
select * from employees  
where ename = 'John'
```

Ex2: Create Index emp-name-job-date-idx
On employees (ename, job, hiredate)
Compute statistics;

```
→ select * from employees  
where ename = 'John'  
AND hiredate = '12/06/08'  
AND job = 'Manager';
```

(2) DROP

Ex3: Create Unique Index emp-name-idx
On employees (ename)

creating
(For unique index, the records
in the column-name should
be unique) ELSE Error

(2) DROP INDEX

Ex1: Drop Index emp-job-idx

Compute statistics: (Best way possible)
searches

It collects data for database on columns on which indexes are created and create a statistics in the database.

It ^{is a} makes a smarter way to execute searches/queries.

To update a already created INDEX:

Alter index emp-name-idx
Rebuild Compute statistics;