

# DEPARTMENT OF COMPUTER SCIENCE

## Topic :SQL

### EMP

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm.	Deptno
7839	King	President		17-nov-81	5000		10
7698	Blake	Manager	7839	01-may-81	2850		30
7782	Clark	Manager	7839	09-jun-81	2450		10
7566	Jones	Manager	7839	02-apr-81	2975		20
7654	Martin	salesman	7698	28-sep-81	1250	1400	30
7499	Allen	Salesman	7698	20-feb-81	1600	300	30
7844	Turner	Salesman	7698	08-sep-81	1500	0	30
7900	James	Clerk	7698	03-dec-81	950		30
7521	Ward	Salesman	7698	22-feb-81	1250	500	30
7902	Ford	Analyst	7566	03-dec-81	3000		20
7369	Smith	Clerk	7902	17-dec-80	800		20
7788	Scott	Analyst	7566	09-dec-82	3000		20
7876	Adams	Clerk	7788	12-jan-83	1100		20
7934	Miller	Clerk	7782	23-jan-82	1300		10

### DEPT

Deptno	Deptname	Loc
10	Accounting	New york
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

SQL : SQL is a tool for organizing , managing and retrieving data stored by a computer data base.

### SQL DATA TYPES :

DATA TYPE	DESCRIPTION
NUMBER	Includes just the digits 0 through 9, a decimal points, and a minus sign, if necessary. NUMBER literals are not enclosed in single quotation marks
CHAR or CHARACTER	Fixed –length Character strings
DATE	Calendar Date e.g. mm/dd/yy; dd-mm-yyyy

## SQL COMMANDS

### DATA DEFINITION LANGUAGE COMMANDS

The Data Definition Language (DDL) is used to create and destroy databases and database objects. These commands will primarily be used by database administrators during the setup and removal phases of a database project

### DATA MANIPULATION LANGUAGE COMMANDS(DML)

The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database.

#### I. DATA DEFINITION LANGUAGE COMMANDS

i. CREATE TABLE ii. CREATE VIEW iii.. ALTER TABLE iv DROP TABLE

#### 1.CREATE TABLE COMMAND

**PURPOSE :** Creates an empty table – a table with no rows

**SYNTAX :** **COM1(a)**

```
CREATE TABLE table-name
    ( col-name data-type (size),
      col-name data-type (size),
      ... );
```

**NOTE :** 1. The meaning of size argument varies with the data type.  
If you omit it, your system will Assign a value automatically.

Eg.

**CREATE TABLE XIID**

```
( ADNO NUMBER,  
  NAME CHAR(20),  
  AVG NUMBER  
);
```

### Try u r self.

- i. create a table with the under mentioned structure (table name is emp)
  - a. empno NUMBER(4)
  - b. empname CHAR(10)
  - c. job CHAR(10)
  - d. manager NUMBER(4)
  - e. hiredate DATE
  - f. salary NUMBER(7,2)
  - g. commission NUMBER(7,2)
  - h. Deptno NUMBER(2)

### 2. ALTERING TABLE ONCE IT HAS BEEN CREATED

- a. Adding a new column.

**PURPOSE :** The new column will be added with NULL values for all rows currently in the table. This column will be the last column of the table.

**SYNTAX:**

```
COM2 ALTER TABLE table-name  
      ADD ( col-name data-type (size),  
          col-name data-type (size),  
          ...  
          );
```

e.g. ALTER TABLE XIID  
 ADD sex char(2);

### Try u r self.

- i. To alter the structure of table EMP by adding a column spouses name
- ii. To alter a table structure DEPT by adding a column noofstaff, phno

### 3. Dropping a table

**PURPOSE :** To drop a table. Once this command is given , the table name no longer recognized & no more commands can be given on that object.

**SYNTAX:**

```
COM3 DROP TABLE table -name;
```

e.g. DROP TABLE XIID

---

### DATA MANIPULATION COMMANDS

#### 1. ENTERING VALUES : INSERT Command

```
COM4 SYNTAX:  
      INSERT INTO table-name  
      VALUES ( value1, value2, ...);
```

Eg. INSERT INTO XIID  
 VALUES ( 1001, 'PRIYA',56.7);

**NOTE :** Each value enumerated in the values class must match , the data type of the column into which it is being inserted.

#### 2. INSERTING NULL VALUES :

```
COM5 e.g. INSERT INTO XIID  
      VALUES ( 1002, 'MOTIKA', NULL);
```

#### 3. NAMING COLUMN FOR INSERT:

**PURPOSE:** You can also specify the column you wish to insert the value into by name. This allows you to insert the value in any order.

```
COM6 SYNTAX:  
      INSERT INTO table name ( col-name1, col-name2, col-name3....)  
      VALUES ( value1, value2, value 3.. );
```

e.g. INSERT INTO XIID ( AVG,NAME, ADNO)  
 VALUES( 89.9,'PRIYA',11);

---

- Try u r self:** i. insert a new employee with your own data  
ii. insert a new employee with your own data and comm. as NULL  
iii. Insert a record with suitable data in the table EMP, taking system date as the hiredate  
iv. insert a new row in DEPT table in the following order: loc, deptno, deptname
- 

## 2. REMOVING ROWS FROM THE TABLE

### 1. Removing all the rows from the table

**COM7** SYNTAX: DELETE FROM table – name ;  
e.g. DELETE FROM XIID

The above command deletes all rows from table XIID

---

### 2. Removing only particular rows :

**COM8** SYNTAX: DELETE FROM table – name  
WHERE col-name = exp;

e.g. DELETE FROM XIID  
WHERE ADNO= 1001;

The above command deletes 1001's row from the XIID table.

---

## Try u r self

---

### 4. DELETE command

- i. delete all salesman ii. delete all employees who has no commission  
iii. delete all clerks whose salary is more than 1000
- 

## CHANGING FIELD VALUES : UPDATE COMMAND

**PURPOSE:** Modifies the values of one or more columns in all rows of a table or in selected rows of a table.

### 1. UPDATING ALL THE ROWS

**COM9** SYNTAX:  
UPDATE table –name  
SET col-name = exp;

e.g. UPDATE XIID  
SET AVG = 67;

The above command updates AVG as 67 for all XIID

---

### 2. UPDATING ONLY CERTAIN ROWS :

**COM10** SYNTAX:  
UPDATE table –name  
SET col-name = exp  
WHERE search cond.

e.g. UPDATE XIID  
SET AVG = 78  
WHERE ADNO = 1001;

The above command updates AVG as .78 only for 1001

### UPDATING MULTIPLE COLUMNS:

---

**COM1** SYNTAX:

UPDATE table –name  
SET col-name = exp, col-name = exp....  
WHERE search cond;

e.g. UPDATE XIID  
SET AVG = 45, NAME = 'XYZ'  
WHERE SNUM = 1001;

## USING EXPRESSIONS IN UPDATE

E.G. increase all salespeople commission by double  
**COM12** UPDATE XIID  
SET COMM = COMM \* 2 ;

### 3. UPDATING TO NULL

e.g. **COM13** UPDATE XIID  
SET AVG = NULL  
WHERE SNUM = 1004;

## TRY UR SELF

### UPDATE command

- i. Increase all salary by 120
- ii. increase salary of all clerks by 20% of salary
- iii. increase sal of managers by 12% of salary whose salary is more than 2500
- iv. increase salary of all salesman by 10% of salary and commission by 10
- v. increase salary of analyst by 100
- vi. change blake job as vice president

## Using SQL to retrieve Information from Tables

### The SELECT command

#### I. Selecting everything in easy way

To display all employees

SELECT \* FROM EMP; NOTE : \* used to display all column of the table

#### II Looking only a certain columns of a table

To display name and salary of the all the employees

SELECT ENAME, SAL FROM EMP;

#### III. COLUMN REORDING

SELECT SAL, ENAME, JOB ,HIREDATE, EMPNO, COMM, DEPTNO  
FROM EMP;

#### IV. ELIMINATING REDUNDANT DATA

DISTINCT is an argument that provides a way for you to eliminate duplicate values from u r SELECT command

eg. To diaplay all different jobs

SELECT DISTINCT JOB FROM EMP

Note : the above command will display all jobs without duplicates

## QUALIFIED SELECTION – THE WHERE CLAUSE

The SQL queries that retrieves all rows of a table are useful for data base browsing & reports , but for little else. Usually you will want to select only some rows of a table and include only these rows in the query results . The WHERE clause is used to specify the rows you want to retrieve.

SYNTAX:

SELECT col-names or \*  
FROM table – name  
WHERE search cond.

For each row search cond. Will produce one of these results:

If S.C. is TRUE : Then the row is included in query results

If S.C. is FALSE : Then the row is excluded in query results

If S.C is NULL : Then the row is excluded in query results

### SEARCH CONDITIONS

- I. COMPARISION TEST
- II. RANGE TEST
- III. SET MEMBERSHIP TEST
- IV. PATTERN MATCHING TEST

I. COMPARISION TEST: Compares the value of one exp. To values of other exp.

Operators : = (equal), <>( Not Equal), >, <, >=, <= ( similar to C++)

**TRY UR SELF**

- i. Display all salesman and their salary
  - ii. Display name of the all managers
  - iii. Display whose is getting salary 2500
  - iv. Display all employees who has commission
  - v. Display all employees who has no commission
  - vi. Display all employees who earns sal more than
  - vii. Display those who are not working as manager
  - viii. Display all employees who earn salary more than salary
- 
- ix. Display all employees who earns 2000 and more than 2000

2. BOOLEAN OPERATORS : 1. AND 2. OR 3. NOT

AND

- i. Display all managers who earn more than 5000
- ii. Display all sales man who joins after 01-apr-98
- iii. Display all mangers who has commission and getting sal more than 3000
- iv. Display all employees whos not as salesman getting sal more than 1000

OR

- i. Display all managers and clerks
- ii. Display all managers or salesman whose earns sal more than 1000
- iii. Display all employees working ass clerks or hired after 12-may-99

NOT

- i. Display all managers those who are nor getting sal more than 2000
- ii. Write a command that exclude salesman whose is earning more than 1000

3.SET MEMBERSHIP TEST IN OPERATOR

- I. To display all managers ,salesman, clerks
- II. To display all employees except managers, clerks
- III. display employees who join april 3<sup>rd</sup> or 4 th

4. RANGE TEST (BETWEEN AND)

Note : BETWEEN operator includes both the boundaries

- I. Display all managers who earn 2000 to 7000
- II. Display all managers who earn 2000 to 7000  
(in this 2000 and 7000 should not be included)
- iii. display employees who join april 3<sup>rd</sup> or 4 th

BETWEEN with characters ;u can use it to select range from alphabetical ordering.

Note when you use BETWEEN to extract alphabetical ranges , you will usually go one letter beyond the last letter you want to include , or add a Z after your second boundary

- i. Display employees whose names begin with a letter A to G
- ii. Display employees whose names not begin with a letter A to G

5. PATTERN MATCHING TEST (like operator)

LIKE can be applied only to fields of type CHAR

Wildcards:

1. % stands for sequence of any number of characters(including zero characters)
  2. \_ stands for any single character .
- 
- i. To display all employees whose name start with 'A'
  - ii. To display all employees whose name end with 'A'
  - iii. To display all employees whose name start with 'A' and end with 'S'
  - iv. To display all employees whose name has LL in it
  - v. To display all employees whose name start with 'A' or 'T'
  - vi. Display all employees whose names include either of substring "TH" or "LL"
  - vii. To display all employees whose name 's third letter 'A'
  - viii. To display all employees whose name has four letter only
  - ix. To display all employees whose name start with 'A' and third letter as 'w' and end with S
  - x. To find all those employees whose job does not start with "M"

SUMMARIZING DATA WITH AGGREGATE FUNCTIONS

1. SUM() : computes total of a column ; If DISTINCT option is specified then the duplicates values are ignored.

SYNTAX: SELECT SUM( col-name)  
FROM table - name;

Eg. SELECT SUM ( SAL )  
FROM EMP Ans:

SYNTAX : SELECT SUM( DISTINCT col-name)  
FROM table - name;

Eg. SELECT SUM ( DISTINCT SAL )  
FROM EMP Ans:

---

2. AVG () : computes the average value in column; If DISTINCT option is specified then the duplicates values are ignored.

SYNTAX: SELECT AVG( col-name)  
FROM table - name;

Eg. SELECT AVG ( SAL )  
FROM EMP; Ans:

SYNTAX : SELECT AVG( DISTINCT col-name)  
FROM table - name;

Eg. SELECT AVG ( DISTINCT SAL )  
FROM EMP Ans:

---

3. MIN() : Finds the smallest value in the column

SYNTAX: SELECT MIN( col-name)  
FROM table - name;

Eg. SELECT MIN ( SAL )  
FROM EMP Ans:

---

4. MAX () : finds the largest value in a column

SYNTAX: SELECT MAX( col-name)  
FROM table - name;

Eg. SELECT MAX ( MAX )  
FROM EMP; Ans:

---

5. i. COUNT () Counts the number of values in a column[ excludes NULLs ]  
ii. COUNT(\*) Counts rows of query results[ includes NULLs ]

1. To count no. of values(different) in a column

SYNTAX1: SELECT COUNT ( col-name)  
FROM table -name;

E.G. SELECT COUNT (COMM) [ excludes NULLs ]  
FROM EMP; Ans :

---

SYNTAX2: SELECT COUNT ( DISTINCTcol-name)  
FROM table -name;

E.G. SELECT COUNT ( DISTINCT JOB)[ excludes NULLs ]  
FROM EMP; Ans :

---

### **TRY UR SELF :**

#### **SUM()**

- I. To display total salary paid to clerks
- II. To display total salary paid to managers without duplicates

#### **AVG()**

- I. To display average salary paid to clerks
- II. To display average salary paid to managers without duplicates

#### **MAX()**

- I. To display highest salary of all managers
- II. To display highest salary of all managers joined after 5<sup>th</sup> April
- III. To display difference between highest and lowest salaries

#### **Min()**

- I. To display highest salary of all managers
- II. To display highest salary of all managers joined after 5<sup>th</sup> April

#### **Count()**

- i. To display number employees getting commission
- ii. To display number employees not getting commission
- iii. To display all jobs without duplicates
- iv. To display number of employees working as managers and clerks
- v. To display cardinality of the table

### **FORMATTING QUERY OUTPUT**

#### **i. expressions in select command**

- i. To display employs names, salary , job and annual salary of all employees
- ii. To display employs names, salary , job and annual salary of all clerks
- iii. To display employs names, salary , job and total salary(sal+comm)of all employees

#### **ii. Strings in select command**

- i. Display all employees in following format  
Eg. Clark is working as manager
- ii. Clark is earning 2450\$ every month

#### **iii. Ordering output by fields (ORDER BY) COMMAND**

- i. To arrange whole table alphabetical order of name
- ii. Display ename. Job , hiredate in descending order of hiredate
- iii. To arrange all employees ascending order of salary
- iv. To arrange all employees reverse order of name
- v. To arrange all employees reverse order of salary and alphabetical order of name
- vi. To arrange whole table alphabetical order of name using column number

### **SUMMARIZING DATA WITH AGGREGATE FUNCTIONS&**

#### **group by and having clause**

- i. To display highest salary of managers and clerks
- ii. To display highest salary in each job
- iii. Display the sum , average, highest , lowest salary of all employees
- iv. Display the sum , average, highest , lowest salary of all employees in job wise
- v. Display the number of employees with same job;
- vi. Display number employees joined in each hiredate
- vii. To display number of employees working as manager and clerks separately
- viii. Display the manager number and salary of the lowest paid employee for that manager. Exclude any one where the manager id is not known , excludes any groups where minimum salary is less than rs. 1000
- ix. Show the average salary of all departments with more than 3 people for a job

x. Show the total salary of all departments with less than 3 people for a job

## QUERYING MULTIPLE TABLES AT ONCE

I. Display name, salary and department name of all employees

II. Display name of the employee working in London

III. Display name, job, department name and location of all accounting managers

IV. Display all details of sales manager

V. Display name, salary and department name of all employees who earn commission

RELATION: STOCK

ITNO	ITNAME	BRDNAME	COUNTRY	QTY	UPRICE
1	CPU	INTEL	US	1200	2000
2	CPU	MOTROLA	JAPAN	4000	1450
3	MONIOR	LG	JAPAN	2000	1200
4	MOUSE	MICROSOFT	US	75000	100
5	HARD DISK	WESTRNDIGITAL	US	79000	450
6	CPU	IBM	US	65000	13000
7	HARDDISK	SEGATE	TAIWAN	76800	600
8	RAM	SONY	JAPAN	450000	9000

1. CREATE TABLE command

Q1. Write create table command for the above table

Q2. Create a STOCK table so that all ITNO values are different from one another

2. ALTER TABLE COMMAND

\_\_Q1. Add a new column named 'city'

Q2. Add a new columns named phno, pd

3. INSERT INTO command

Q1. Insert a new item in the above table with u r own dada

Q2. Insert a new item with NULL value in the qty field

Q3. Insert new item in the following order : ITNAME, UPRICE, COUNTRY, QTY, ITNO, BRDNAME

UPDATE command

Q1. Increase the price of the all STOCKS by 20% of the current price

Q2. Increase QTY of all SONY STOCK

Q3. change all CPU's country as NULL

Q4. Change all JAPAN 's STOCK as INDIA and increase the price by 78%

Q5. Increase the price of all JAPAN STOCK with 50%

Q6. To change all HARD DISKS country as UK

5. DELETE command

Q1. Delete all STOCK in the above table

Q2. Delete all item from SONY

Q3. Delete all STOCK having price more than 12000

4. SELECT command

Q1. To display all STOCK

Q2. To display all brand names of STOCK

Q3. To show all STOCK having price more than 50000

Q4. To list all STOCK sorted by price in descending order

Q5. To display all Indian STOCK

Q6. To display all STOCK which are not produced from JAPAN

Q7. To find the minimum price of the STOCK

Q8. To find the highest price of an STOCK

Q9. To count no. STOCK produced from JAPAN

Q10. To total value of all STOCK

Q11. To find Total value of all STOCK produced from US

Q12. To find average price of all US STOCK

Q15. To display a report listing ITNO, ITNAME, COUNTRY, COMMISSION (1.2% OF UPRICE) of STOCK