

Handout 2

SQL Statements and their syntax

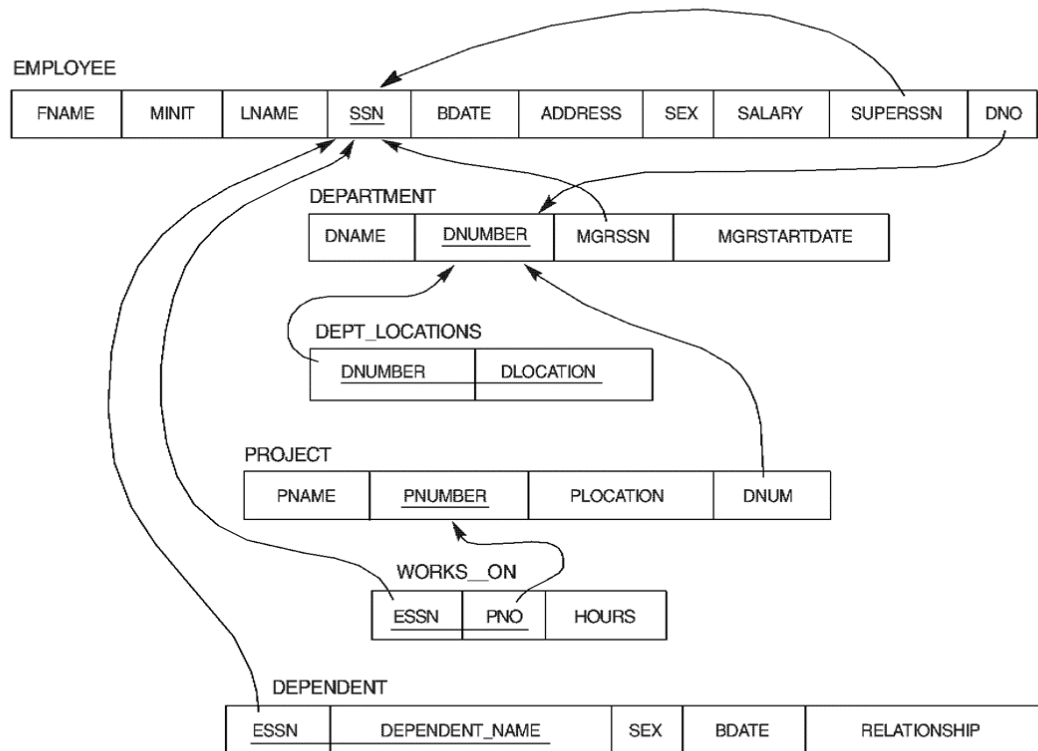
1. **SELECT** *column_name,column_name* **FROM** *table_name*;
 a. select basic from employee;
 2. **SELECT** *column_name, column_name* **FROM** *table_name* **ORDER BY** *column_name* **ASC|DESC, column_name ASC|DESC**;
 a. select ename from employee order by basic DESC;
 3. **UPDATE** *table_name* **SET** *column1=value1,column2=value2,...* **WHERE** *some_column=some_value*;
 a. UPDATE employee SET basic =15000, ename= 'partha' WHERE emp_no='100';
 4. **DELETE FROM** *table_name* **WHERE** *some_column=some_value*;
 a. DELETE FROM employee WHERE ename='partha';
 5. **SELECT** *column_name(s)* **FROM** *table1* **JOIN** *table2* **ON** *table1.column_name=table2.column_name*;
 a. SELECT name FROM candidate JOIN job ON candidate.candidate_id=job.candidate_id; (this query means two tables candidate and job have the same attribute candidate_id)
 6. **SELECT** *column_name(s)* **FROM** *table_name* **WHERE** *column_name* **BETWEEN** *value1* **AND** *value2*;
 a. SELECT basic, ename FROM employee WHERE basic BETWEEN 10000 AND 19000;
 7. **SELECT** *column_name(s)* **FROM** *table_name* **WHERE** *column_name* **IN** (*value1,value2,...*);
 a. SELECT basic, ename FROM employee WHERE ename IN ('Anil', 'Ganesh');
 8. **SELECT COUNT(column_name)** **FROM** *table_name*;
 a. SELECT COUNT(ename) FROM employee;
 9. **SELECT** *column_name, aggregate_function(column_name)* **FROM** *table_name* **WHERE** *column_name* **operator** *value* **GROUP BY** *column_name* **HAVING** *aggregate_function(column_name)* **operator** *value*;
 10. **SELECT** *column_name* [, *column_name*] **FROM** *table1* [, *table2*] **WHERE** *column_name* **OPERATOR** (**SELECT** *column_name* [, *column_name*] **FROM** *table1* [, *table2*] [**WHERE**])
- **SELECT basic FROM employee WHERE ename LIKE 'A%'; (to retrieve names starting with A; retrieve names ending with A is like 'A%'; retrieve names have A anywhere in the name 'A%')**

11. Create tables as per the following diagram using entity and referential integrity keys.

- **SQL Query to create a table with primary key:**
 - create table candidate(candidate_id varchar(5) primary key, name varchar(30) unique, telephone number);
- **SQL Query to create a table with a foreign key:**
 - create table job(position_id varchar(5) primary key, candidate_id varchar(5), date2 date, foreign key(candidate_id) references candidate(candidate_id));

Create a database associated to a project with the given below six tables. Use referential integrity and entity integrity concepts and create the tables in a specific order as per your understanding. Also insert the data according to the second diagram.

Figure 7.7 Referential integrity constraints displayed on the COMPANY relational database schema diagram.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

1. Consider the following tables.

WORKS (Pname, Cname, Salary)

LIVES(Pname, Street, City)

LOCATED_IN(Cname, City)

Where Pname = Person name, Cname=Company name and Mgrname = Manager name.

Execute the SQL queries for the following:

- a. Create the tables and assign the appropriate primary key and foreign key constraints.
- b. Insert at least 5 tuples for each table.
- c. List the names of the people who work for the company Wipro along with the cities they live in.
- d. Find the people who work for the company 'Infosys' with a salary more than Rs. 50000/-. List the names of the people along with the street and city addresses.
- e. Find the names of the persons who live and work in the same city.
- f. Find the names of the persons who do not work for 'Infosys'.
- g. Find the persons whose salaries are more than that of all of the 'oracle' employees.

- h. Find the names of the companies that are located in every city where the company 'Infosys' is located.
- i. Display the average salary and number of the employees company wise in descending order of salary.
- j. Delete the record of a particular employee from the LIVES.
- k. Display the details of the employee whose name has the alphabet 'k'.