Expt. 4 3 September 2012

**SQL Constraints**

**Aim:**

To study constraints in SQL and solve the given queries.

**Theory:**

Constraints are used to limit the type of data that can go into a table. They can be specified when a table is created (with the CREATE TABLE statement) or after the table is created (with the ALTER TABLE statement).

Following are commonly used constraints:

* NOT NULL

The NOT NULL constraint enforces a column to NOT accept NULL values. The NOT NULL constraint enforces a field to always contain a value. This means that you cannot insert a new record, or update a record without adding a value to this field.

* UNIQUE

The UNIQUE constraint uniquely identifies each record in a database table. The UNIQUE and PRIMARY KEY constraints both provide a guarantee for uniqueness for a column or set of columns. A PRIMARY KEY constraint automatically has a UNIQUE constraint defined on it.

* PRIMARY KEY

The PRIMARY KEY constraint uniquely identifies each record in a database table. Primary keys must contain unique values. A primary key column cannot contain NULL values. Each table should have a primary key, and each table can have only ONE primary key.

* FOREIGN KEY

A FOREIGN KEY in one table points to a PRIMARY KEY in another table. The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables. The FOREIGN KEY constraint also prevents that invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the table it points to.

* CHECK

The CHECK constraint is used to limit the value range that can be placed in a column. If you define a CHECK constraint on a single column it allows only certain values for this column. If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

* DEFAULT

The DEFAULT constraint is used to insert a default value into a column. The default value will be added to all new records, if no other value is specified.

**Queries:**

***1) For each employee, retrieve employee's first name and last name with salary and first name,*** ***last name and salary of his/her immediate supervisor***

> select e.fname, e.minit, e.lname, e.salary, x.fname as s\_fname, x.minit as s\_minit, x.lname as s\_lname, x.salary as s\_salary from employee as e join employee as x on e.superssn=x.ssn

+----------+-------+---------+--------+----------+---------+---------+----------+

| fname | minit | lname | salary | s\_fname | s\_minit | s\_lname | s\_salary |

+----------+-------+---------+--------+----------+---------+---------+----------+

| John | B | Smith | 30000 | Franklin | T | Wong | 40000 |

| Franklin | T | Wong | 40000 | James | E | Borg | 55000 |

| Joyce | A | English | 25000 | Franklin | T | Wong | 40000 |

| Ramesh | K | Narayan | 38000 | Franklin | T | Wong | 40000 |

| Jennifer | S | Wallace | 43000 | James | E | Borg | 55000 |

| Ahmad | V | Jabbar | 25000 | Jennifer | S | Wallace | 43000 |

| Alicia | J | Zelaya | 25000 | Jennifer | S | Wallace | 43000 |

+----------+-------+---------+--------+----------+---------+---------+----------+

***2) For a project that has greater than 3 employees working on it, list project name, number of employees, average salary, managers name and controling department name***

> select a.pname,d.dname,m\_fname,m\_lname,emp\_count,e.avg\_salary from project as a,

(select pnumber,count(pnumber) as emp\_count from project,works\_on where pno=pnumber group by pnumber having count(pnumber)>2) as b,

(select dname,pname,pnumber from project,department where dnumber=dnum) as c,

(select dname, fname as m\_fname, lname as m\_lname from employee, department where mgrssn=ssn) as d,

(select pno,avg(salary) as avg\_salary from employee,works\_on where ssn=essn group by pno) as e

where a.pnumber=b.pnumber and b.pnumber=c.pnumber and c.dname=d.dname and e.pno=b.pnumber

+-----------------+----------------+----------+---------+-----------+------------+

| pname | dname | m\_fname | m\_lname | emp\_count | avg\_salary |

+-----------------+----------------+----------+---------+-----------+------------+

| ProductY | Research | Franklin | Wong | 3 | 31666.6667 |

| Computerization | Administration | Jennifer | Wallace | 3 | 30000.0000 |

| Reorganization | Headquarters | James | Borg | 3 | 46000.0000 |

| Newbenefits | Administration | Jennifer | Wallace | 3 | 31000.0000 |

+-----------------+----------------+----------+---------+-----------+------------+

***3) Find list of all employees who were born on 15th of any month in 1960's***

> select fname,lname from employee where bdate like '196\_-\_\_-15'

+--------+---------+

| fname | lname |

+--------+---------+

| Ramesh | Narayan |

+--------+---------+

***4) For each employee retrieve his name, address, dept. name for which he works, and the name of his/her dependent. If the employee do not have a dependent, represent it as null. (Write the query using (i)union (ii)outer joins)***

> select fname,minit,lname,address,dname,dependentname from (select ssn,fname,minit,lname,address,dname from employee join department on dno=dnumber) as x left outer join dependent on ssn=essn;

+----------+-------+---------+--------------------------+----------------+---------------+

| fname | minit | lname | address | dname | dependentname |

+----------+-------+---------+--------------------------+----------------+---------------+

| John | B | Smith | 731 Fondren, Houston, TX | Research | Alice |

| John | B | Smith | 731 Fondren, Houston, TX | Research | Elizabeth |

| John | B | Smith | 731 Fondren, Houston, TX | Research | Micheal |

| Franklin | T | Wong | 638 Voss, Houston, TX | Research | Alice |

| Franklin | T | Wong | 638 Voss, Houston, TX | Research | Joy |

| Franklin | T | Wong | 638 Voss, Houston, TX | Research | Theodore |

| Joyce | A | English | 5631 Rice, Houston, TX | Research | NULL |

| Ramesh | K | Narayan | 975 Fire Oak, Humble, TX | Research | NULL |

| James | E | Borg | 450 Stone, Houston, TX | Headquarters | NULL |

| Jennifer | S | Wallace | 291 Berry, Bellaire, TX | Administration | Abner |

| Ahmad | V | Jabbar | 980 Dallas, Houston, TX | Administration | NULL |

| Alicia | J | Zelaya | 3321 Castle Spring , TX | Administration | NULL |

+----------+-------+---------+--------------------------+----------------+---------------+

***5) Make a list of project numbers for projects that involve an employee whose last name is Smith either as a worker or as a manager of the department that controls the project (use joins).***

> select pnumber from ( (select pnumber,lname from employee, works\_on, project where ssn=essn and pno=pnumber and dno=dnum)union (select pnumber,lname from employee, department, project where mgrssn=ssn and dnum=dno)) as z where z.lname='Smith';

+---------+

| pnumber |

+---------+

| 1 |

| 2 |

+---------+

***6) Create a new table 'dept\_info' with 3 fields: dept\_name, num\_employees, total\_salary. Insert into the table name of each department, the total number of employees in each department and the total salary earned by all the employees of that department.***

> create table dept\_info (select dname, count(dno) as num\_employees, sum(salary) as total\_salary from employee join department on dno=dnumber group by dname);

> select \* from dept\_info;

+----------------+---------------+--------------+

| dname | num\_employees | total\_salary |

+----------------+---------------+--------------+

| Administration | 3 | 93000 |

| Headquarters | 1 | 55000 |

| Research | 4 | 133000 |

+----------------+---------------+--------------+

***7) Give all the employees of Research department a 10% hike.***

> update employee set salary=salary-0.1\*salary where dno=5

***8) Create a view to store following information:***

***\* Name of the employee***

***\* Age***

***\* Address***

***\* Name of the project for which he works***

***\* Number of working hours***

***Display the information present in the view.***

> create view temp\_view as (select fname,minit,lname,bdate,address,pname,hours from employee,works\_on,project where essn=ssn and pno=pnumber);

***9) Update the view by making John Smith work on 'ProductY' rather than 'ProductX'. Check the resulting updates in the base table. Rewrite the updates to the base tables. Drop the newly created view.***

> update temp\_view set pname='ProductY' where fname='John' and lname='Smith';

> select \* from temp\_view;

+----------+-------+---------+------------+--------------------------+-----------------+-------+

| fname | minit | lname | bdate | address | pname | hours |

+----------+-------+---------+------------+--------------------------+-----------------+-------+

| John | B | Smith | 1965-01-09 | 731 Fondren, Houston, TX | ProductY | 32.5 |

| Joyce | A | English | 1972-07-31 | 5631 Rice, Houston, TX | ProductX | 20 |

| John | B | Smith | 1965-01-09 | 731 Fondren, Houston, TX | ProductY | 7.5 |

| Franklin | T | Wong | 1955-12-08 | 638 Voss, Houston, TX | ProductY | 10 |

| Joyce | A | English | 1972-07-31 | 5631 Rice, Houston, TX | ProductY | 20 |

| Franklin | T | Wong | 1955-12-08 | 638 Voss, Houston, TX | ProductZ | 10 |

| Ramesh | K | Narayan | 1962-09-15 | 975 Fire Oak, Humble, TX | ProductZ | 40 |

| Franklin | T | Wong | 1955-12-08 | 638 Voss, Houston, TX | Computerization | 10 |

| Ahmad | V | Jabbar | 1969-03-29 | 980 Dallas, Houston, TX | Computerization | 35 |

| Alicia | J | Zelaya | 1968-07-19 | 3321 Castle Spring , TX | Computerization | 10 |

| Franklin | T | Wong | 1955-12-08 | 638 Voss, Houston, TX | Reorganization | 10 |

| James | E | Borg | 1937-11-10 | 450 Stone, Houston, TX | Reorganization | NULL |

| Jennifer | S | Wallace | 1941-06-20 | 291 Berry, Bellaire, TX | Reorganization | 15 |

| Jennifer | S | Wallace | 1941-06-20 | 291 Berry, Bellaire, TX | Newbenefits | 20 |

| Ahmad | V | Jabbar | 1969-03-29 | 980 Dallas, Houston, TX | Newbenefits | 5 |

| Alicia | J | Zelaya | 1968-07-19 | 3321 Castle Spring , TX | Newbenefits | 30 |

+----------+-------+---------+------------+--------------------------+-----------------+-------+

> select fname,minit,lname,bdate,address,pname,hours from employee,works\_on,project where essn=ssn and pno=pnumber;

+----------+-------+---------+------------+--------------------------+-----------------+-------+

| fname | minit | lname | bdate | address | pname | hours |

+----------+-------+---------+------------+--------------------------+-----------------+-------+

| John | B | Smith | 1965-01-09 | 731 Fondren, Houston, TX | ProductX | 32.5 |

| Joyce | A | English | 1972-07-31 | 5631 Rice, Houston, TX | ProductX | 20 |

| John | B | Smith | 1965-01-09 | 731 Fondren, Houston, TX | ProductY | 7.5 |

| Franklin | T | Wong | 1955-12-08 | 638 Voss, Houston, TX | ProductY | 10 |

| Joyce | A | English | 1972-07-31 | 5631 Rice, Houston, TX | ProductY | 20 |

| Franklin | T | Wong | 1955-12-08 | 638 Voss, Houston, TX | ProductZ | 10 |

| Ramesh | K | Narayan | 1962-09-15 | 975 Fire Oak, Humble, TX | ProductZ | 40 |

| Franklin | T | Wong | 1955-12-08 | 638 Voss, Houston, TX | Computerization | 10 |

| Ahmad | V | Jabbar | 1969-03-29 | 980 Dallas, Houston, TX | Computerization | 35 |

| Alicia | J | Zelaya | 1968-07-19 | 3321 Castle Spring , TX | Computerization | 10 |

| Franklin | T | Wong | 1955-12-08 | 638 Voss, Houston, TX | Reorganization | 10 |

| James | E | Borg | 1937-11-10 | 450 Stone, Houston, TX | Reorganization | NULL |

| Jennifer | S | Wallace | 1941-06-20 | 291 Berry, Bellaire, TX | Reorganization | 15 |

| Jennifer | S | Wallace | 1941-06-20 | 291 Berry, Bellaire, TX | Newbenefits | 20 |

| Ahmad | V | Jabbar | 1969-03-29 | 980 Dallas, Houston, TX | Newbenefits | 5 |

| Alicia | J | Zelaya | 1968-07-19 | 3321 Castle Spring , TX | Newbenefits | 30 |

+----------+-------+---------+------------+--------------------------+-----------------+-------+

> drop view temp\_view;

***10) On the newly created table 'dept\_info' add the following constraints.***

***\* Make dept\_name as primary key.***

> alter table dept\_info add constraint dept\_pkey primary key (dname);

***\* Put a check constraint whether number of employees should be greater than 2.***

> alter table dept\_info add constraint empnum\_chk check (num\_employees>2);

***\* Add a new column 'manager\_name' with a NOT NULL constraint.***

> alter table dept\_info add column manager\_name varchar(32) not null;

***\* Update the table to insert manager's name.***

> update dept\_info set manager\_name=(select fname from employee join depart

ment on mgrssn=ssn where dname='Research') where dname=' Research';

> update dept\_info set manager\_name=(select fname from employee join depart

ment on mgrssn=ssn where dname='Administration') where dname='Administration';

> update dept\_info set manager\_name=(select fname from employee join depart

ment on mgrssn=ssn where dname=' Headquarters') where dname='Headquarters';

***\* Alter the table to include a constraint on manager name the constraint being he should be a valid employee of the company (foreign key constraint).***

> alter table dept\_info add constraint mgrname\_fkey foreign key (manager\_name) references employee (fname);

***\* Delete a tuple referring to Research dept. in the newly created table.***

> delete from dept\_info where dname='Research';

***\* Drop the manager\_name column from newly created table.***

> alter table dept\_info drop manager\_name;

**Conclusion:**

Constraints in SQL were studied and given queries were solved.