

## DATA CONSTRAINTS AND VIEWS

**AIM:**

To study about various data constraints and views in SQL.

**QUESTION:**

1. Create the following tables with given constraints
  - a. Create a table named Subjects with the given attributes
    - \* Subid (Should not be NULL)
    - \* Subname (Should not be NULL)

Populate the database. Make sure that all constraints are working properly.

SUB_ID	SUB_NAME
1	Maths
2	Physics
3	Chemistry
4	English

- i) Alter the table to set subid as the primary key.

```
hishamalip : psql — Konsole
CREATE TABLE
asd_lab=# INSERT INTO subjects VALUES(1,'maths'),
                                         (2, 'physics'),
                                         (3, 'chemistry'),
                                         (4, 'english');

INSERT 0 4
asd_lab=# SELECT * FROM subjects;
 subid | subname
-----+-----
      1 | maths
      2 | physics
      3 | chemistry
      4 | english
(4 rows)

asd_lab=# ALTER TABLE subjects
asd_lab=#         ADD PRIMARY KEY (subid);
ALTER TABLE
asd_lab=#
```

b. Create a table named Staff with the given attributes

- staffid (Should be UNIQUE)
- staffname
- dept
- Age (Greater than 22)
- Salary (Less than 35000)

Populate the database. Make sure that all constraints are working properly.

STAFF_ID	STAFF_NAME	DEPT	AGE	SALARY
1	John	Purchasing	24	30000
2	Sera	Sales	25	20000
3	Jane	Sales	28	25000

```
asd_lab=# CREATE TABLE staff(staffid INT UNIQUE,
                                staffname TEXT,
                                dept TEXT,
                                age INT CHECK (age > 22),
                                salary INT CHECK (salary < 35000));

CREATE TABLE
asd_lab=# INSERT INTO staff
asd_lab=# VALUES (1, 'John', 'Purchasing', 24, 30000),
asd_lab=#          (2, 'Sera', 'Sales', 25, 20000),
asd_lab=#          (3, 'Jane', 'Sales', 28, 25000);
INSERT 0 3
asd_lab=# SELECT * FROM staff;
 staffid | staffname | dept      | age | salary
-----+-----+-----+----+-----
      1 | John     | Purchasing | 24  | 30000
      2 | Sera     | Sales     | 25  | 20000
      3 | Jane     | Sales     | 28  | 25000
(3 rows)
```

i) Delete the check constraint imposed on the attribute salary

ii) Delete the unique constraint on the attribute staffid

```
asd_lab=# \d staff;
          Table "public.staff"
  Column | Type  | Collation | Nullable | Default
-----+-----+-----+-----+-----
 staffid | integer |           |          |
 staffname | text   |           |          |
  dept    | text   |           |          |
   age    | integer |           |          |
  salary  | integer |           |          |
Indexes:
    "staff_staffid_key" UNIQUE CONSTRAINT, btree (staffid)
Check constraints:
    "staff_age_check" CHECK (age > 22)
    "staff_salary_check" CHECK (salary < 35000)

asd_lab=# ALTER TABLE staff DROP CONSTRAINT staff_age_check;
ALTER TABLE
asd_lab=# ALTER TABLE staff DROP CONSTRAINT staff_s;
staff_salary_check staff_staffid_key
asd_lab=# ALTER TABLE staff DROP CONSTRAINT staff_salary_check;
ALTER TABLE
asd_lab=# \d staff;
          Table "public.staff"
  Column | Type  | Collation | Nullable | Default
-----+-----+-----+-----+-----
 staffid | integer |           |          |
 staffname | text   |           |          |
  dept    | text   |           |          |
   age    | integer |           |          |
  salary  | integer |           |          |
Indexes:
    "staff_staffid_key" UNIQUE CONSTRAINT, btree (staffid)

asd_lab=#
```

- c. Create a table named Bank with the following attributes
- bankcode (To be set as Primary Key, type = varchar(3) )
  - bankname (Should not be NULL)
  - headoffice
  - branches (Integer value greater than Zero)

Populate the database. Make sure that all constraints are working properly. All constraints have to be set after creating the table.

BANKCODE	BANK NAME	HEADOFFICE	BRANCHOFFICE
AAA	SIB	Ernakulam	6
BBB	Federal	Kottayam	5
CCC	Canara	Trivandrum	3

```

asd_lab=# CREATE TABLE bank(bankcode VARCHAR(3),
                             bankname VARCHAR(20),
                             headoffice VARCHAR(20),
                             branches INT);
CREATE TABLE
asd_lab=# ALTER TABLE bank
          ALTER COLUMN bankname SET NOT NULL;
ALTER TABLE
asd_lab=# ALTER TABLE bank ADD PRIMARY KEY(bankcode);
ALTER TABLE
asd_lab=# ALTER TABLE bank ADD CONSTRAINT branches CHECK(branches > 0);
ALTER TABLE
asd_lab=# \d bank
               Table "public.bank"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 bankcode     | character varying(3) |           | not null |
 bankname     | character varying(20) |           | not null |
 headoffice   | character varying(20) |           |          |
 branches     | integer              |           |          |
Indexes:
    "bank_pkey" PRIMARY KEY, btree (bankcode)
Check constraints:
    "branches" CHECK (branches > 0)
asd_lab=#

```

```

asd_lab=# INSERT INTO bank VALUES('AAA', 'SIB', 'Ernakulam', 6),
                                   ('BBB', 'Federal', 'Kottayam', 5),
                                   ('CCC', 'Canara', 'Trivandrum', 3);
INSERT 0 3
asd_lab=# SELECT * FROM bank ;
 bankcode | bankname | headoffice | branches
-----+-----+-----+-----
 AAA      | SIB      | Ernakulam  |        6
 BBB      | Federal  | Kottayam   |        5
 CCC      | Canara   | Trivandrum |        3
(3 rows)

```

d. Create a table named Branch with the following attributes

- branchid (To be set as Primary Key)
- branchname (Set Default value as 'New Delhi')
- bankid (Foreign Key:- Refers to bank code of Bank table)

i) Populate the database. Make sure that all constraints are working properly.

ii) During database population, demonstrate how the DEFAULT Constraint is satisfied.

BRANCH_ID	BRANCHNAME	BANKID
01	Kottayam	CCC
02	New Delhi	AAA

i)

```
asd_lab=# CREATE TABLE branch(
           branchid INT PRIMARY KEY,
           branchname TEXT DEFAULT 'New Delhi',
           bankid CHAR(3) REFERENCES bank(bankcode));
CREATE TABLE
asd_lab=# \d branch
Table "public.branch"
  Column      |      Type      | Collation | Nullable |      Default
-----+-----+-----+-----+-----
 branchid    | integer        |           | not null |
 branchname  | text           |           |          | 'New Delhi'::text
 bankid      | character(3)   |           |          |
Indexes:
    "branch_pkey" PRIMARY KEY, btree (branchid)
Foreign-key constraints:
    "branch_bankid_fkey" FOREIGN KEY (bankid) REFERENCES bank(bankcode)
asd_lab=#
```

ii)

```
asd_lab=# INSERT INTO branch VALUES(01, 'Kottayam', 'CCC');
INSERT 0 1
asd_lab=# INSERT INTO branch(branchid, bankid) VALUES(02, 'AAA');
INSERT 0 1
asd_lab=# SELECT * FROM branch;
 branchid | branchname | bankid
-----+-----+-----
        1 | Kottayam   | CCC
        2 | New Delhi  | AAA
(2 rows)
asd_lab=#
```

- iii) Delete the bank with bank code 'SBT' and make sure that the corresponding entries are getting deleted from the related tables.

BANKCODE	BANK NAME	HEADOFFICE	Branch Office
AAA	SIB	Ernakulam	6
BBB	Federal	Kottayam	5
CCC	Canara	Trivandrum	3
SBT	Indian	Delhi	7

Branch I D	Branch Name	Bank ID
1	Kottayam	CCC
2	New Delhi	AAA
5	Calicut	SBT

```
asd_lab=# INSERT INTO bank VALUES('SBT', 'Indian', 'Delhi', 7);
INSERT 0 1
asd_lab=# INSERT INTO branch VALUES(5, 'Calicut', 'SBT');
INSERT 0 1
asd_lab=# SELECT * FROM bank ;
 bankcode | bankname | headoffice | branches
-----+-----+-----+-----
 AAA      | SIB      | Eranakulam |         6
 BBB      | Federal  | Kottayam   |         5
 CCC      | Canara   | Trivandrum |         3
 SBT      | Indian   | Delhi      |         7
(4 rows)

asd_lab=# SELECT * FROM branch;
 branchid | branchname | bankid
-----+-----+-----
        1 | Kottayam   | CCC
        2 | New Delhi  | AAA
        5 | Calicut    | SBT
(3 rows)

asd_lab=# DELETE FROM bank WHERE bankcode = 'SBT';
ERROR:  update or delete on table "bank" violates foreign key constraint "branch_bankid_
fkey" on table "branch"
DETAIL:  Key (bankcode)=(SBT) is still referenced from table "branch".
```

When trying to delete bankcode = 'SBT' from "bank" table, we got an error because it is still referenced from table "branch". We can't delete a foreign key if it still references another table.

To make the corresponding entries are getting deleted from the related tables we need to add 'ON DELETE CASCADE' to the existing foreign key. In order to do this, drop the existing constraint and recreate it with addition of the 'ON DELETE' clause.

```
asd_lab=# ALTER TABLE branch
asd_lab=# DROP CONSTRAINT branch_bankid_fkey;
ALTER TABLE
asd_lab=#
asd_lab=# ALTER TABLE branch
asd_lab=# ADD CONSTRAINT branch_bankid_fkey
asd_lab=# FOREIGN KEY (bankid)
asd_lab=# REFERENCES bank (bankcode)
asd_lab=# ON DELETE CASCADE;
ALTER TABLE
asd_lab=#
```

After deletion

```
asd_lab=# DELETE FROM bank WHERE bankcode = 'SBT';
DELETE 1
asd_lab=# SELECT * FROM bank;
 bankcode | bankname | headoffice | branches
-----+-----+-----+-----
AAA      | SIB      | Eranakulam |        6
BBB      | Federal  | Kottayam   |        5
CCC      | Canara   | Trivandrum |        3
(3 rows)

asd_lab=# SELECT * FROM branch;
 branchid | branchname | bankid
-----+-----+-----
        1 | Kottayam   | CCC
        2 | New Delhi  | AAA
(2 rows)
```

iv) Drop the Primary Key using ALTER command

```
asd_lab=# \d branch
Table "public.branch"
  Column      | Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
branchid     | integer   |           | not null |
branchname   | text      |           |          | 'New Delhi'::text
bankid       | character(3) |           |          |
Indexes:
    "branch_pkey" PRIMARY KEY, btree (branchid)
Foreign-key constraints:
    "branch_bankid_fkey" FOREIGN KEY (bankid) REFERENCES bank(bankcode) ON DELETE CASCADE

asd_lab=# ALTER TABLE branch DROP CONSTRAINT branch_pkey ;
ALTER TABLE
asd_lab=# \d branch
Table "public.branch"
  Column      | Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
branchid     | integer   |           | not null |
branchname   | text      |           |          | 'New Delhi'::text
bankid       | character(3) |           |          |
Foreign-key constraints:
    "branch_bankid_fkey" FOREIGN KEY (bankid) REFERENCES bank(bankcode) ON DELETE CASCADE

asd_lab=#
```

2. Create a View named sales\_staff to hold the details of all staff working in sales Department

```
asd_lab=# CREATE VIEW sales_staff AS
asd_lab=# SELECT * FROM staff
asd_lab=# WHERE dept = 'Sales';
CREATE VIEW
asd_lab=# SELECT * FROM sales_staff;
 staffid | staffname | dept | age | salary
-----+-----+-----+-----+-----
      2 | Sera     | Sales | 25 | 20000
      3 | Jane     | Sales | 28 | 25000
(2 rows)
```

3. Drop table branch. Create another table named branch and name all the constraints as given below:

Constraint name	Column	Constraint
Pk	branch_id	Primary key
Df	branch_name	Default : 'New Delhi'
Fk	bankid	Foreign key/References

```
asd_lab=# CREATE TABLE branch(
    branch_id INT CONSTRAINT Pk PRIMARY KEY,
    branch_name varchar(20) CONSTRAINT Df DEFAULT 'New Delhi',
    bankid char(3) CONSTRAINT Fk REFERENCES bank(bankcode) ON DELETE CASCADE
);
CREATE TABLE
asd_lab=# \d branch
      Table "public.branch"
  Column      |      Type      | Collation | Nullable |      Default
-----+-----+-----+-----+-----
 branch_id   | integer        |           | not null |
 branch_name | character varying(20) |           |          | 'New Delhi'::character varying
  bankid     | character(3)   |           |          |
Indexes:
    "pk" PRIMARY KEY, btree (branch_id)
Foreign-key constraints:
    "fk" FOREIGN KEY (bankid) REFERENCES bank(bankcode) ON DELETE CASCADE
asd_lab=#
```

- Delete the default constraint in the table
- Delete the primary key constraint

```
asd_lab=# ALTER TABLE branch
asd_lab=# ALTER COLUMN branch_name DROP DEFAULT;
ALTER TABLE
asd_lab=#
asd_lab=# ALTER TABLE branch DROP CONSTRAINT Pk;
ALTER TABLE
asd_lab=# \d branch
      Table "public.branch"
  Column      |      Type      | Collation | Nullable |      Default
-----+-----+-----+-----+-----
 branch_id   | integer        |           | not null |
 branch_name | character varying(20) |           |          |
  bankid     | character(3)   |           |          |
Foreign-key constraints:
    "fk" FOREIGN KEY (bankid) REFERENCES bank(bankcode) ON DELETE CASCADE
asd_lab=#
```

4. Update the view sales\_staff to include the details of staff belonging to sales department whose salary is greater than 20000.

```
asd_lab=# CREATE OR REPLACE VIEW sales_staff AS
                                SELECT * FROM staff
                                WHERE dept = 'Sales'
                                AND salary > 20000;

CREATE VIEW
asd_lab=# SELECT * FROM sales_staff ;
 staffid | staffname | dept | age | salary
-----+-----+-----+-----+-----
      3 | Jane      | Sales |  28 | 25000
(1 row)

asd_lab=#
```

5. Delete the view sales\_staff.

```
asd_lab=# DROP VIEW sales_staff ;
DROP VIEW
asd_lab=# SELECT * FROM sales_staff ;
ERROR:  relation "sales_staff" does not exist
LINE 1: SELECT * FROM sales_staff ;
                        ^

asd_lab=#
```

## RESULT:

The query was executed and the output was obtained.