VIT | Vidyalankar Institute of Technology
Accredited A+ by NAAC

DEPARTMENT OF INFORMATION TECHNOLOGY

Name :	Kamraan Mulani
Roll No :	23101A0028
Semester :	IV
Practical no :	10
Title of Practical:	Views and Integrity Constraints
Objective:	To create views and apply various integrity constraints on given datasets and test them.
Date Of Performance:	08-04-2025

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

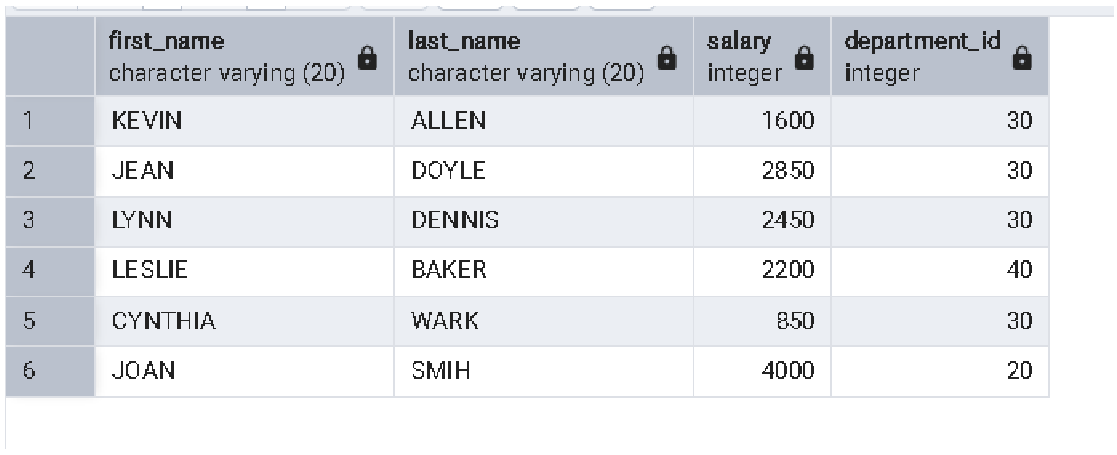
You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.

CREATE VIEW Syntax

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

View for Employee Table

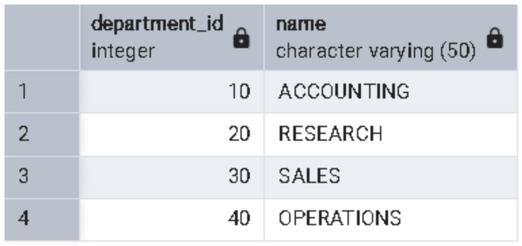
```
create view empview as
select first_name,last_name,salary,department_id from employee_001
where salary>800;

select * from empview;
```



	first_name character varying (20)	last_name character varying (20)	salary integer	department_id integer
1	KEVIN	ALLEN	1600	30
2	JEAN	DOYLE	2850	30
3	LYNN	DENNIS	2450	30
4	LESLIE	BAKER	2200	40
5	CYNTHIA	WARK	850	30
6	JOAN	SMIH	4000	20

View for Department Table

```
create view deptview as
select department_id as , department_001.name  from department_001;
select * from deptview;
```



	department_id integer	name character varying (50)
1	10	ACCOUNTING
2	20	RESEARCH
3	30	SALES
4	40	OPERATIONS

Integrity Constraints in SQL:

The following constraints are commonly used in SQL:

NOT NULL : Ensures that a column cannot have a NULL value.
It guarantees that a field must always contain valid data when a new record is inserted.

UNIQUE : Ensures that all values in a column (or set of columns) are different.
It helps maintain data uniqueness without making it a primary key.

PRIMARY KEY : Combines NOT NULL and UNIQUE to uniquely identify each row in a table.
Each table can have only one primary key, made up of one or more columns.

FOREIGN KEY : Enforces a link between two tables by referencing the primary key of another table.
It ensures referential integrity by restricting invalid data entry.

CHECK : Restricts the values in a column based on a logical condition.
For example, it can ensure that age > 0 or salary <= 100000.

DEFAULT :

Sets a default value for a column when no value is specified during insertion.
This ensures consistency and prevents nulls where defaults are appropriate.

CREATE INDEX : Creates an index on one or more columns to speed up query performance.
It improves data retrieval efficiency but does not affect data integrity directly.

1. Adding Default Constraint to Employee Table :

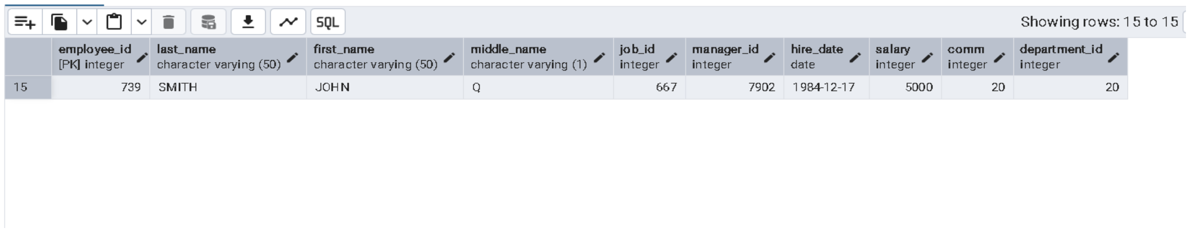
ALTER TABLE employee

ALTER COLUMN salary SET DEFAULT 5000;

INSERT INTO EMPLOYEE VALUES

(739, 'SMITH', 'JOHN', 'Q', 667, 7902, '1984-12-17',DEFAULT, 20, 20);

select * from employee;



Showing rows: 15 to 15

	employee_id [PK] integer	last_name character varying (50)	first_name character varying (50)	middle_name character varying (1)	job_id integer	manager_id integer	hire_date date	salary integer	comm integer	department_id integer
15	739	SMITH	JOHN	Q	667	7902	1984-12-17	5000	20	20

2. Adding Foreign Key Constraint to Department Table :

ALTER TABLE DEPARTMENT

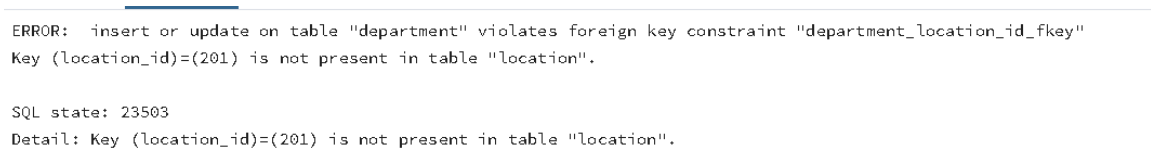
ADD CONSTRAINT fk_department_location

FOREIGN KEY (location_id)

REFERENCES LOCATION(location_id);

INSERT INTO DEPARTMENT (department_id, name,location_id) VALUES

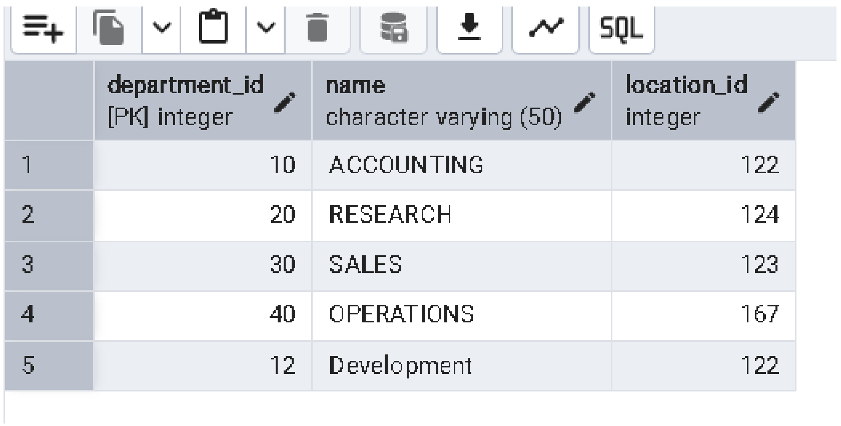
(11, 'ACCOUNTING', 201); -- Inserting a tuple with location id absent in locations table.



```
ERROR:  insert or update on table "department" violates foreign key constraint "department_location_id_fkey"
Key (location_id)=(201) is not present in table "location".

SQL state: 23503
Detail: Key (location_id)=(201) is not present in table "location".
```

INSERT INTO DEPARTMENT (department_id, name,location_id) VALUES

(12, 'Development', 122);

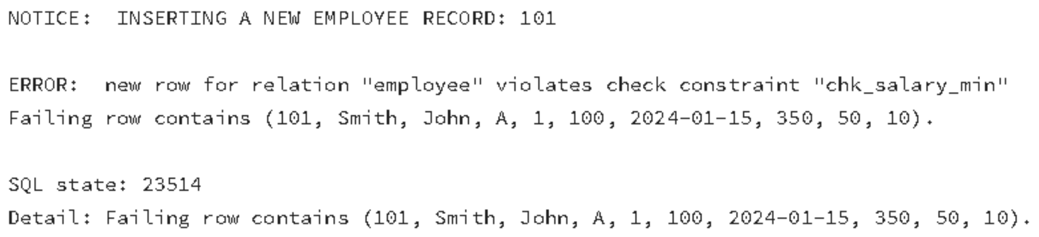


	department_id [PK] integer	name character varying (50)	location_id integer
1	10	ACCOUNTING	122
2	20	RESEARCH	124
3	30	SALES	123
4	40	OPERATIONS	167
5	12	Development	122

3. Adding Check Constraint to Employee Table :

```
ALTER TABLE EMPLOYEE
ADD CONSTRAINT check_salary
CHECK (Salary > 400);
```

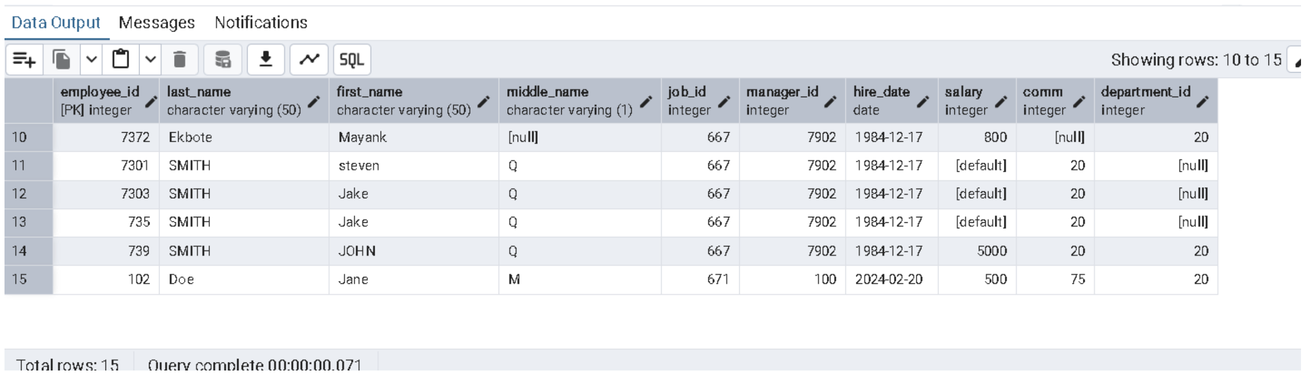
```
INSERT INTO EMPLOYEE (
    Employee_ID, Last_Name, First_Name, Middle_Name, Job_ID,
    Manager_ID, Hire_Date, Salary, Comm, Department_ID)
                              VALUES (
                               101, 'Smith', 'John', 'A', 1,
                               100, '2024-01-15', 350, 50, 10 );
```



```
NOTICE:  INSERTING A NEW EMPLOYEE RECORD: 101

ERROR:  new row for relation "employee" violates check constraint "chk_salary_min"
Failing row contains (101, Smith, John, A, 1, 100, 2024-01-15, 350, 50, 10).

SQL state: 23514
Detail: Failing row contains (101, Smith, John, A, 1, 100, 2024-01-15, 350, 50, 10).
```

```
INSERT INTO EMPLOYEE (
    Employee_ID, Last_Name, First_Name, Middle_Name, Job_ID,
    Manager_ID, Hire_Date, Salary, Comm, Department_ID
) VALUES (
    102, 'Doe', 'Jane', 'M', 671,
    100, '2024-02-20', 500, 75, 20
);
```

select * from employee;


Data Output | Messages | Notifications

Showing rows: 10 to 15

	employee_id [PK] integer	last_name character varying (50)	first_name character varying (50)	middle_name character varying (1)	job_id integer	manager_id integer	hire_date date	salary integer	comm integer	department_id integer
10	7372	Ekbote	Mayank	[null]	667	7902	1984-12-17	800	[null]	20
11	7301	SMITH	steven	Q	667	7902	1984-12-17	[default]	20	[null]
12	7303	SMITH	Jake	Q	667	7902	1984-12-17	[default]	20	[null]
13	735	SMITH	Jake	Q	667	7902	1984-12-17	[default]	20	[null]
14	739	SMITH	JOHN	Q	667	7902	1984-12-17	5000	20	20
15	102	Doe	Jane	M	671	100	2024-02-20	500	75	20

Total rows: 15 | Query complete 00:00:00.071

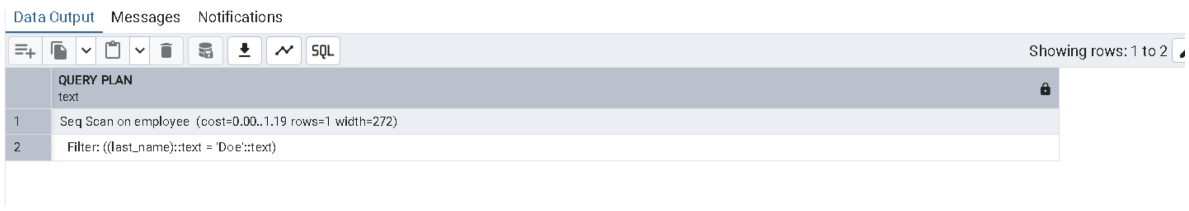
4. Adding Create Index Constraint on Employee Table :

CREATE INDEX idx_employee_lastname

ON EMPLOYEE (Last_Name);

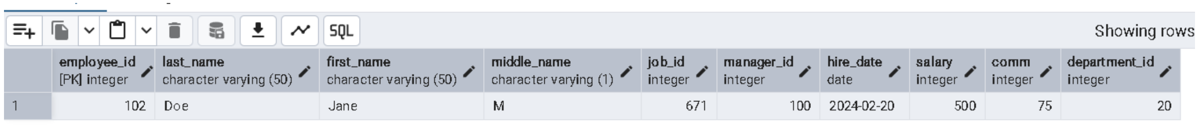
EXPLAIN SELECT * FROM EMPLOYEE WHERE Last_Name = 'Doe';

SELECT * FROM EMPLOYEE WHERE Last_Name = 'Doe'


Data Output | Messages | Notifications

Showing rows: 1 to 2

	QUERY PLAN text
1	Seq Scan on employee (cost=0.00..1.19 rows=1 width=272)
2	Filter: ((last_name)::text = 'Doe'::text)


Showing rows

	employee_id [PK] integer	last_name character varying (50)	first_name character varying (50)	middle_name character varying (1)	job_id integer	manager_id integer	hire_date date	salary integer	comm integer	department_id integer
1	102	Doe	Jane	M	671	100	2024-02-20	500	75	20

5. Adding Not Null constraint on Employee Table :

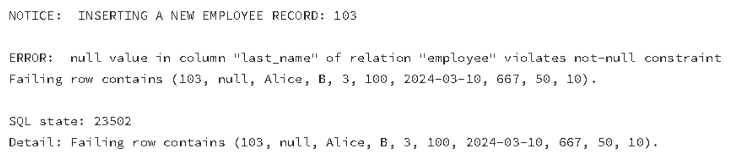
ALTER TABLE EMPLOYEE

ALTER COLUMN Last_Name SET NOT NULL;

INSERT INTO EMPLOYEE

VALUES (103, NULL, 'Alice', 'B', 667,100, '2024-03-10', 900, 50, 10);

-- Inserting with Last_name set as NULL.

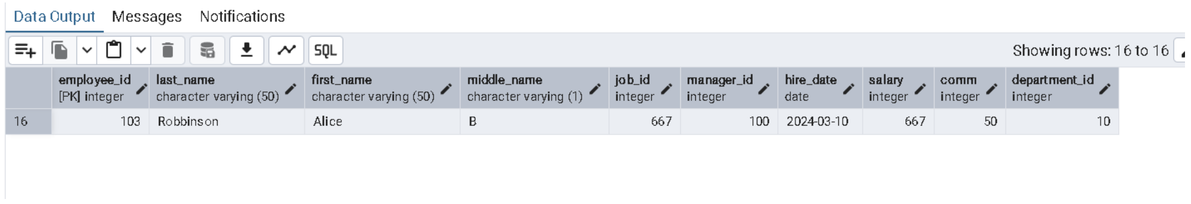


```
NOTICE:  INSERTING A NEW EMPLOYEE RECORD: 103

ERROR:  null value in column "last_name" of relation "employee" violates not-null constraint
Failing row contains (103, null, Alice, B, 3, 100, 2024-03-10, 667, 50, 10).

SQL state: 23502
Detail: Failing row contains (103, null, Alice, B, 3, 100, 2024-03-10, 667, 50, 10).
```

INSERT INTO EMPLOYEE

VALUES (103, NULL, 'Alice', 'B', 667,100, '2024-03-10', 667, 50, 10);

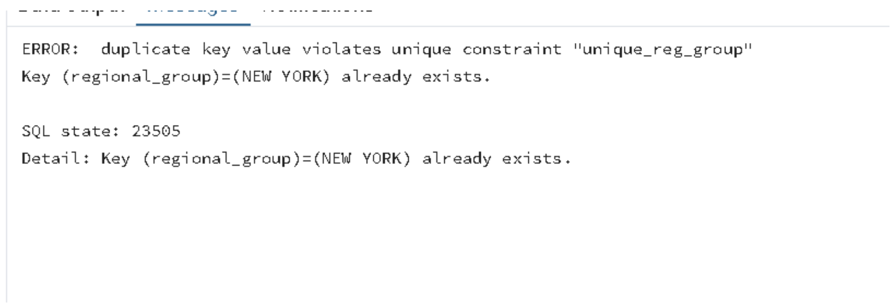
Select * from employee;



	employee_id [PK] integer	last_name character varying (50)	first_name character varying (50)	middle_name character varying (1)	job_id integer	manager_id integer	hire_date date	salary integer	comm integer	department_id integer
16	103	Robbinson	Alice	B	667	100	2024-03-10	667	50	10

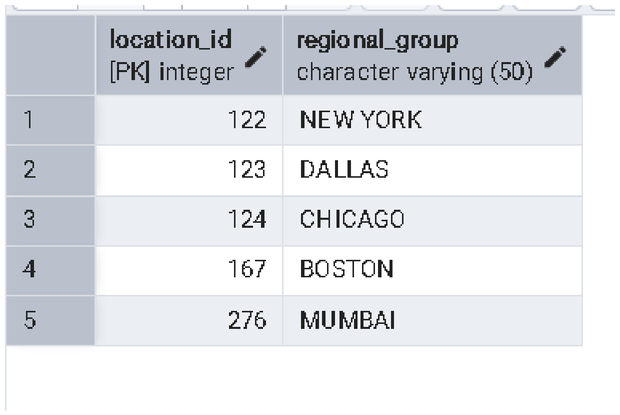
6. **Adding Unique constraint on Employee Table :**

```
ALTER TABLE LOCATION

ADD CONSTRAINT unique_reg_group UNIQUE (Regional_Group);


-- This will fail because 'NEW YORK' already exists in the table

INSERT INTO LOCATION (Location_ID, Regional_Group)

VALUES (168, 'NEW YORK');
```



```
ERROR:  duplicate key value violates unique constraint "unique_reg_group"
Key (regional_group)=(NEW YORK) already exists.

SQL state: 23505
Detail: Key (regional_group)=(NEW YORK) already exists.
```

```
INSERT INTO LOCATION (Location_ID, Regional_Group)

VALUES (276, 'MUMBAI');
```



	location_id [PK] integer	regional_group character varying (50)
1	122	NEW YORK
2	123	DALLAS
3	124	CHICAGO
4	167	BOSTON
5	276	MUMBAI