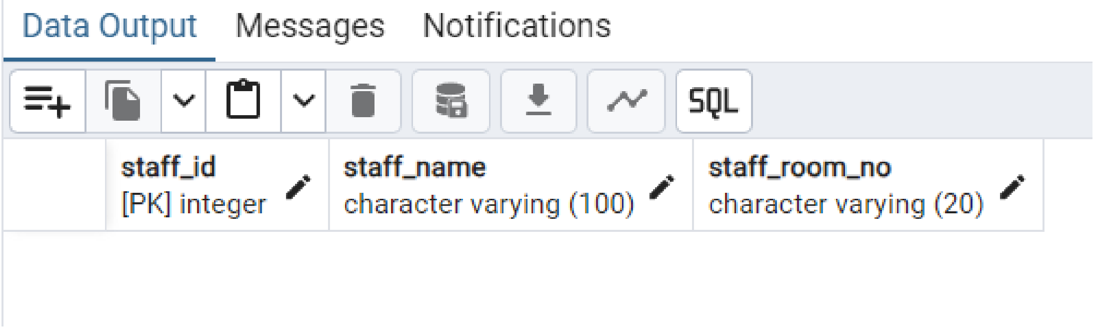
DBMS PROJECT PHASE 4

GROUP 5

Aparna Padma B	AM.EN.U4AIE22005
Anuvind MP	AM.EN.U4AIE22010
R S Harish Kumar	AM.EN.U4AIE22042
Siddharth Menon	AM.EN.U4AIE22048

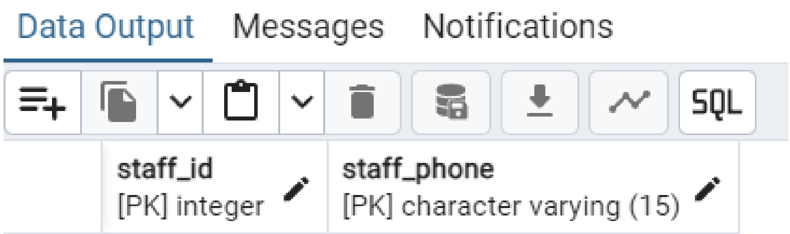
1. Create Tables with necessary constraints pertaining to the relational schema using appropriate DDL statements

```
CREATE TABLE Staff (

  staff_id INT PRIMARY KEY,

  staff_name VARCHAR(100) NOT NULL,

  staff_room_no VARCHAR(20) NOT NULL

);
```



staff_id [PK] integer	staff_name character varying (100)	staff_room_no character varying (20)

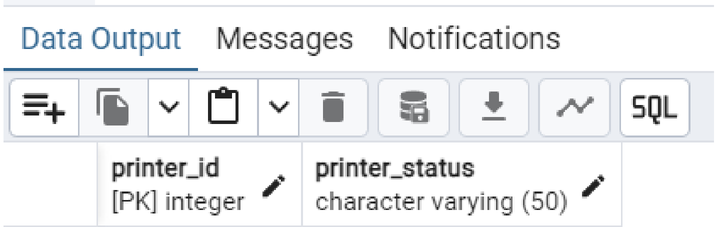
```
CREATE TABLE Staff_Phone (

  staff_id INT,

  staff_phone VARCHAR(15) NOT NULL,

  PRIMARY KEY (staff_id, staff_phone),
    FOREIGN KEY (staff_id) REFERENCES Staff(staff_id) ON DELETE CASCADE
);
```



staff_id
[PK] integer

staff_phone
[PK] character varying (15)

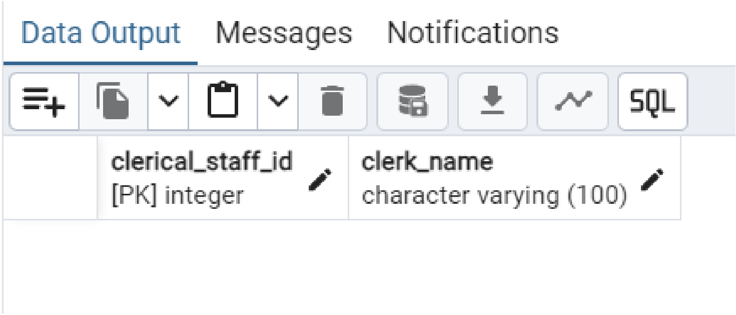
```
CREATE TA
    printer_id INT PRIMARY KEY,
    printer_status VARCHAR(50) NOT NULL
);
```



printer_id
[PK] integer

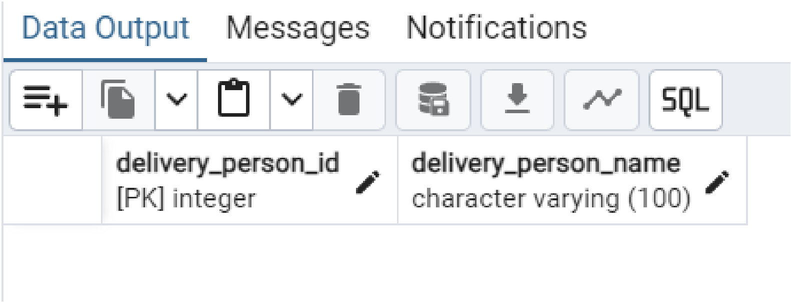
printer_status
character varying (50)

```
CREATE TABLE Clerical_Staff (
    clerical_staff_id INT PRIMARY KEY,
    clerk_name VARCHAR(100) NOT NULL
);
```



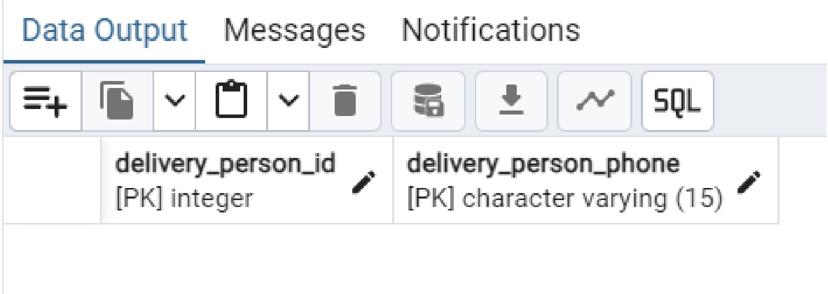
clerical_staff_id [PK] integer	clerk_name character varying (100)

```
CREATE TABLE Delivery_Person (

    delivery_person_id INT PRIMARY KEY,

    delivery_person_name VARCHAR(100) NOT NULL

);
```



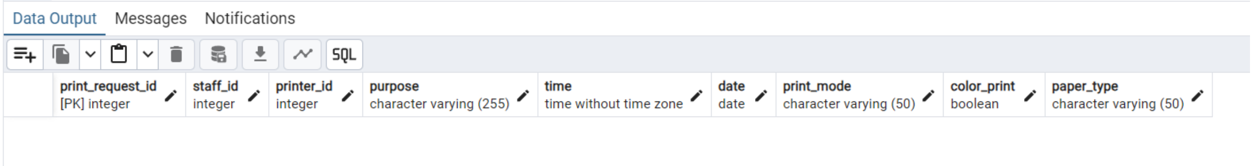
delivery_person_id [PK] integer	delivery_person_name character varying (100)

```
CREATE TABLE Delivery_Phone (

   delivery_person_id INT,

   delivery_person_phone VARCHAR(15) NOT NULL,

   PRIMARY KEY (delivery_person_id, delivery_person_phone),

   FOREIGN KEY (delivery_person_id) REFERENCES Delivery_Person(delivery_person_id) ON
DELETE CASCADE

);
```



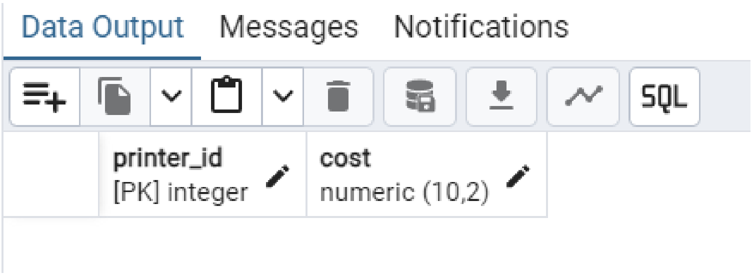
	delivery_person_id [PK] integer	delivery_person_phone [PK] character varying (15)

```
CREATE TABLE Print_Request (
   print_request_id INT PRIMARY KEY,
   staff_id INT,
   printer_id INT,
   purpose VARCHAR(255),
   time TIME NOT NULL,
   date DATE NOT NULL,
   print_mode VARCHAR(50),
   color_print BOOLEAN,
   paper_type VARCHAR(50),
   FOREIGN KEY (staff_id) REFERENCES Staff(staff_id) ON DELETE SET NULL,
   FOREIGN KEY (printer_id) REFERENCES Printer(printer_id) ON DELETE SET NULL
);
```



	print_request_id [PK] integer	staff_id integer	printer_id integer	purpose character varying (255)	time time without time zone	date date	print_mode character varying (50)	color_print boolean	paper_type character varying (50)

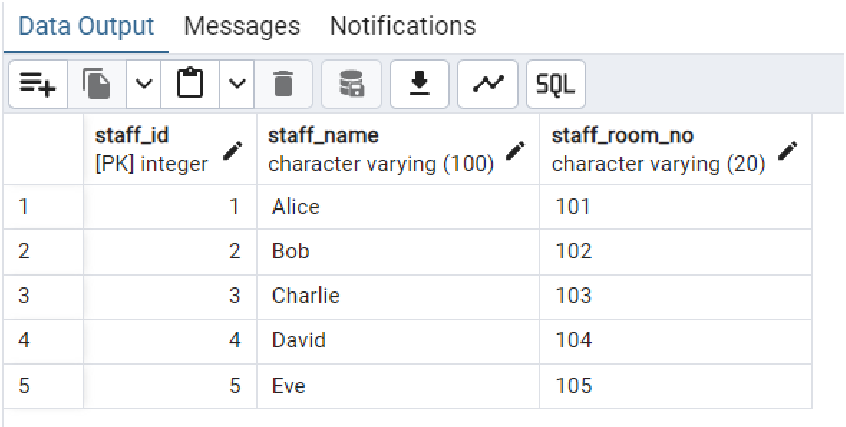
```
CREATE TABLE Printer_Cost (
   printer_id INT PRIMARY KEY,
    cost DECIMAL(10, 2) NOT NULL,

    FOREIGN KEY (printer_id) REFERENCES Printer(printer_id) ON DELETE CASCADE

);
```



	printer_id [PK] integer	cost numeric (10,2)

2. Populate tables with few tuples

```
INSERT INTO Staff VALUES

(1, 'Alice', '101'),

(2, 'Bob', '102'),

(3, 'Charlie', '103'),

(4, 'David', '104'),

(5, 'Eve', '105');
```



	staff_id [PK] integer	staff_name character varying (100)	staff_room_no character varying (20)
1	1	Alice	101
2	2	Bob	102
3	3	Charlie	103
4	4	David	104
5	5	Eve	105

INSERT INTO Staff_Phone VALUES

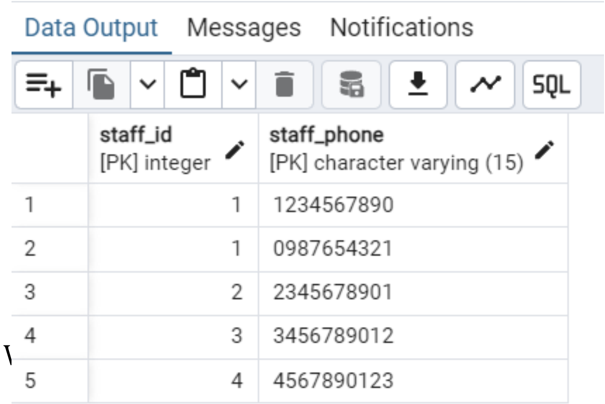
(1, '1234567890'),

(1, '0987654321'),

(2, '2345678901'),

(3, '3456789012'),

(4, '4567890123');



	staff_id [PK] integer	staff_phone [PK] character varying (15)
1	1	1234567890
2	1	0987654321
3	2	2345678901
4	3	3456789012
5	4	4567890123

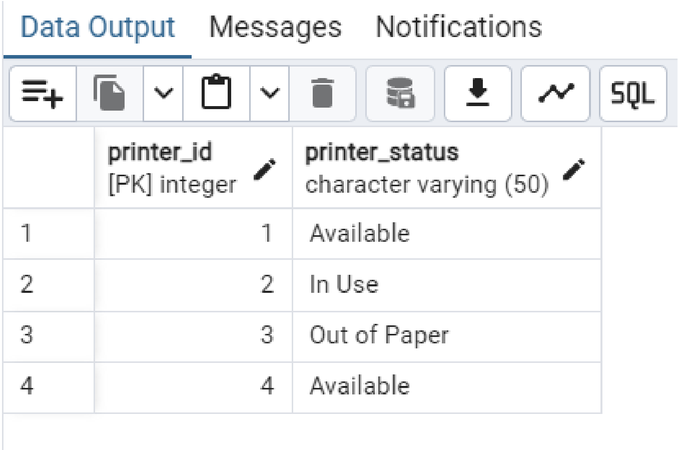
INSERT INTO Printer V

(1, 'Available'),

(2, 'In Use'),

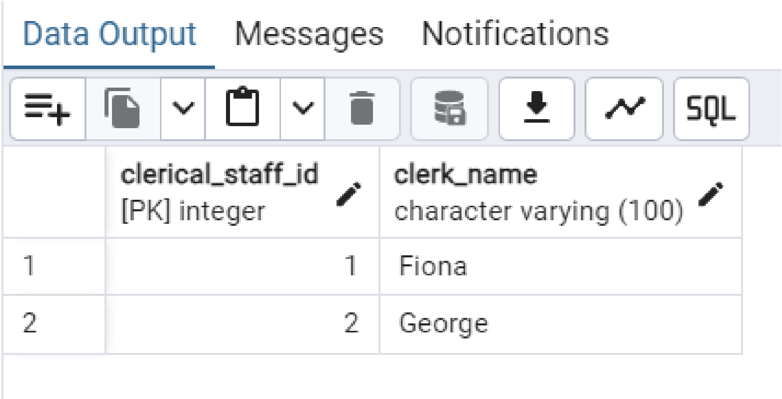
(3, 'Out of Paper'),

(4, 'Available');



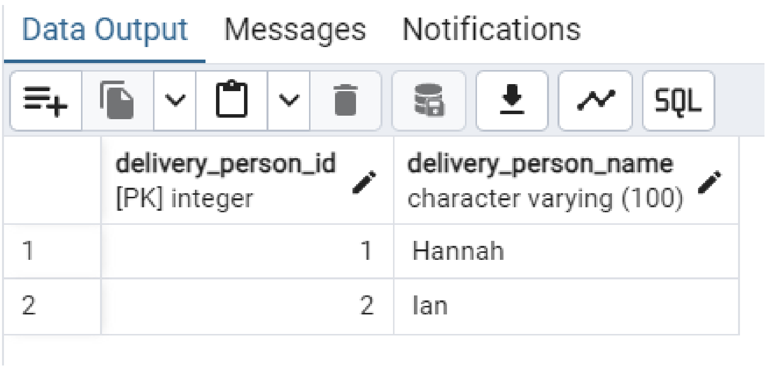
	printer_id [PK] integer	printer_status character varying (50)
1	1	Available
2	2	In Use
3	3	Out of Paper
4	4	Available

```
INSERT INTO Clerical_Staff VALUES

(1, 'Fiona'),

(2, 'George');
```



	clerical_staff_id [PK] integer	clerk_name character varying (100)
1	1	Fiona
2	2	George

```
INSERT INTO Delivery_Person VALUES

(1, 'Hannah'),

(2, 'Ian');
```



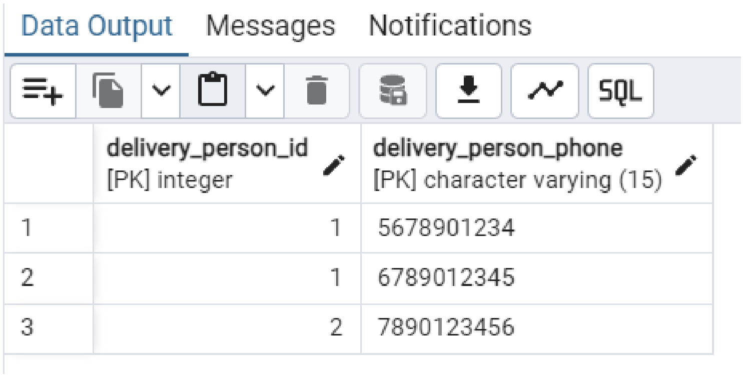
	delivery_person_id [PK] integer	delivery_person_name character varying (100)
1	1	Hannah
2	2	Ian

INSERT INTO Delivery_Phone VALUES

(1, '5678901234'),

(1, '6789012345'),

(2, '7890123456');



	delivery_person_id [PK] integer	delivery_person_phone [PK] character varying (15)
1	1	5678901234
2	1	6789012345
3	2	7890123456

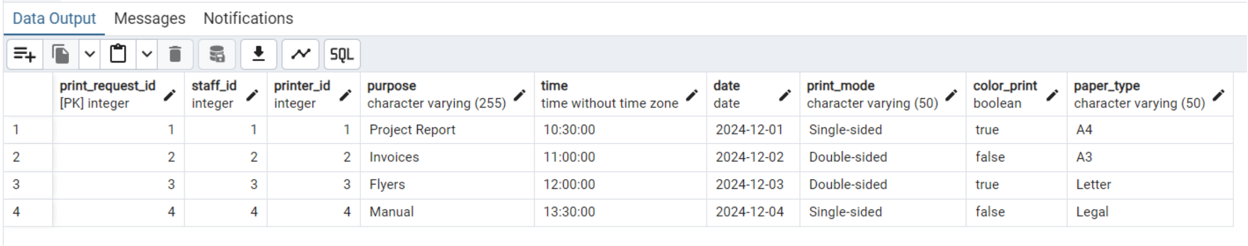
INSERT INTO Print_Request VALUES

(1, 1, 1, 'Project Report', '10:30:00', '2024-12-01', 'Single-sided', TRUE, 'A4'),

(2, 2, 2, 'Invoices', '11:00:00', '2024-12-02', 'Double-sided', FALSE, 'A3'),

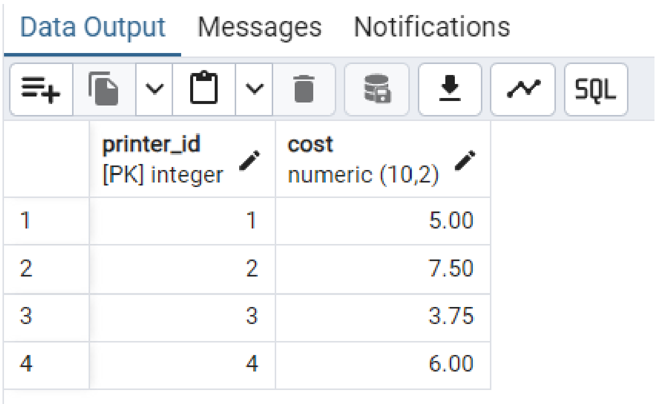
(3, 3, 3, 'Flyers', '12:00:00', '2024-12-03', 'Double-sided', TRUE, 'Letter'),

(4, 4, 4, 'Manual', '13:30:00', '2024-12-04', 'Single-sided', FALSE, 'Legal');



	print_request_id [PK] integer	staff_id integer	printer_id integer	purpose character varying (255)	time time without time zone	date date	print_mode character varying (50)	color_print boolean	paper_type character varying (50)
1	1	1	1	Project Report	10:30:00	2024-12-01	Single-sided	true	A4
2	2	2	2	Invoices	11:00:00	2024-12-02	Double-sided	false	A3
3	3	3	3	Flyers	12:00:00	2024-12-03	Double-sided	true	Letter
4	4	4	4	Manual	13:30:00	2024-12-04	Single-sided	false	Legal

```
INSERT INTO Printer_Cost VALUES

(1, 5.00),

(2, 7.50),

(3, 3.75),

(4, 6.00);
```



	printer_id [PK] integer	cost numeric (10,2)
1	1	5.00
2	2	7.50
3	3	3.75
4	4	6.00

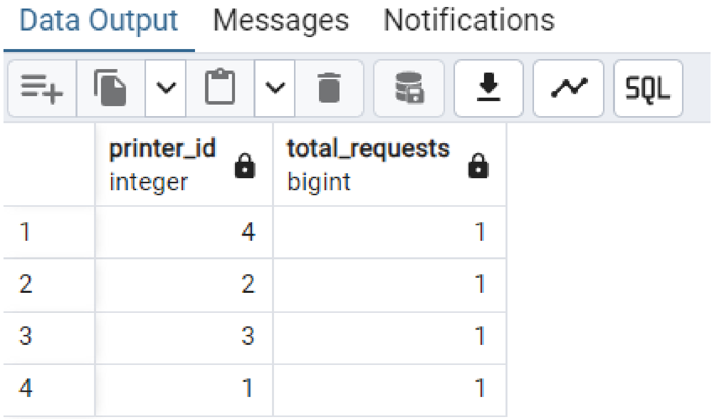
3. Generate 10 queries based on the tables created in (2). The queries must be based on the following:
   i. Group by…having
   ii. Order by
   iii. Join
   iv. Aggregate functions
   v. Query having Boolean operators
   vi. Query having arithmetic operators
   vii. A search query using string operators
   viii. Usage of to_char, extract
   ix. Between, IN, Not between, Not IN
   x. Set operations

1. Group by... having

Use case: Calculate the total number of print requests handled by each printer, filtering out printers with less than 2 requests.

Code:

```
SELECT printer_id, COUNT(*) AS total_requests

FROM Print_Request

GROUP BY printer_id

HAVING COUNT(*) >= 1;
```



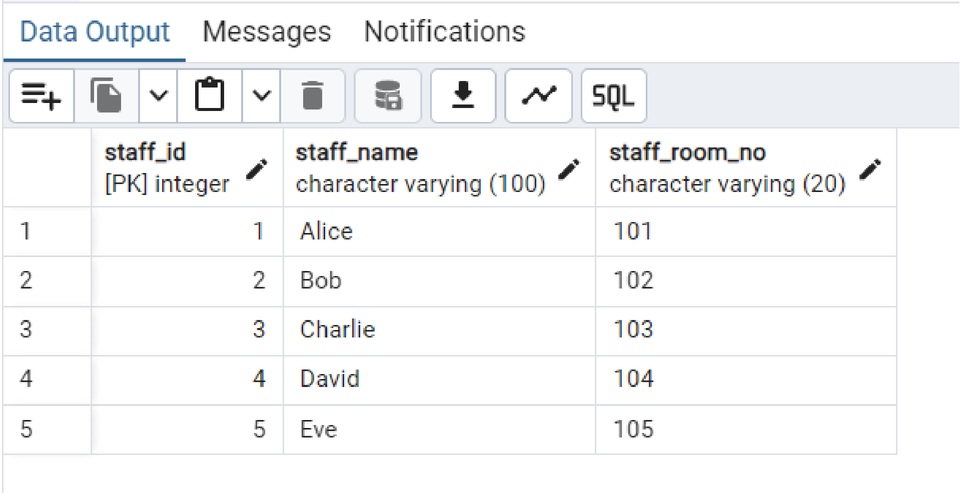
	printer_id integer	total_requests bigint
1	4	1
2	2	1
3	3	1
4	1	1

2. Order by

Use case: List all staff in alphabetical order.

Code:

```
SELECT * FROM Staff

ORDER BY staff_name;
```



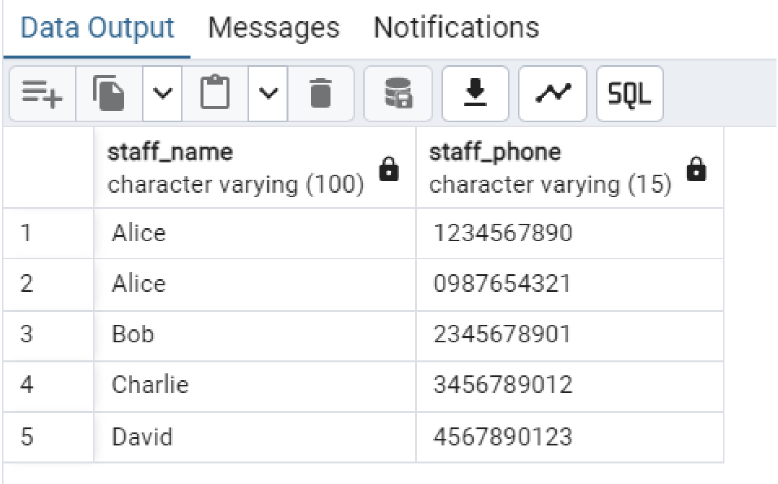
	staff_id [PK] integer	staff_name character varying (100)	staff_room_no character varying (20)
1	1	Alice	101
2	2	Bob	102
3	3	Charlie	103
4	4	David	104
5	5	Eve	105

3. Join

Use case: Find the phone numbers of the staff who made print requests

Code:

```
SELECT s.staff_name, sp.staff_phone

FROM Staff s

JOIN Staff_Phone sp ON s.staff_id = sp.staff_id;
```



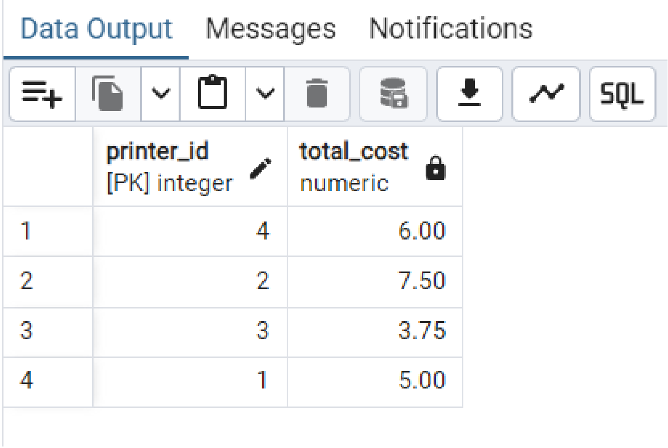
	staff_name character varying (100)	staff_phone character varying (15)
1	Alice	1234567890
2	Alice	0987654321
3	Bob	2345678901
4	Charlie	3456789012
5	David	4567890123

4. Aggregate functions

Use case: Find the total cost incurred for each printer.

Code:

```
SELECT p.printer_id, SUM(pc.cost) AS total_cost

FROM Printer p

JOIN Printer_Cost pc ON p.printer_id = pc.printer_id

GROUP BY p.printer_id;
```



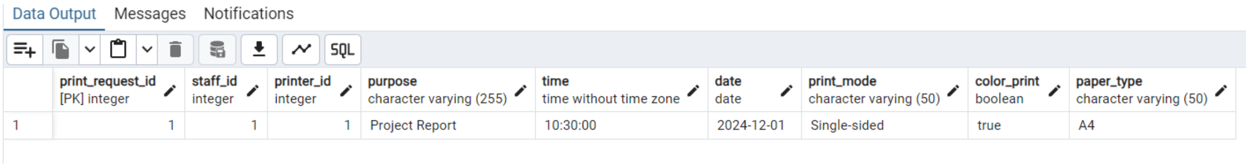
	printer_id [PK] integer	total_cost numeric
1	4	6.00
2	2	7.50
3	3	3.75
4	1	5.00

5. Query having Boolean operators

Use case: Find all color print requests made on A4 paper.

Code:

```
SELECT *

FROM Print_Request

WHERE color_print = TRUE AND paper_type = 'A4';
```



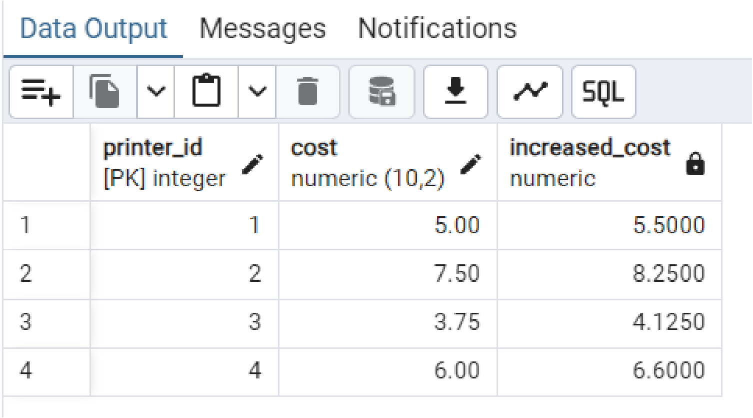
	print_request_id [PK] integer	staff_id integer	printer_id integer	purpose character varying (255)	time time without time zone	date date	print_mode character varying (50)	color_print boolean	paper_type character varying (50)
1	1	1	1	Project Report	10:30:00	2024-12-01	Single-sided	true	A4

6. Query having arithmetic operators

Use case: Increase all printer costs by 10%.

Code:

```
SELECT printer_id, cost, cost * 1.10 AS increased_cost

FROM Printer_Cost;
```



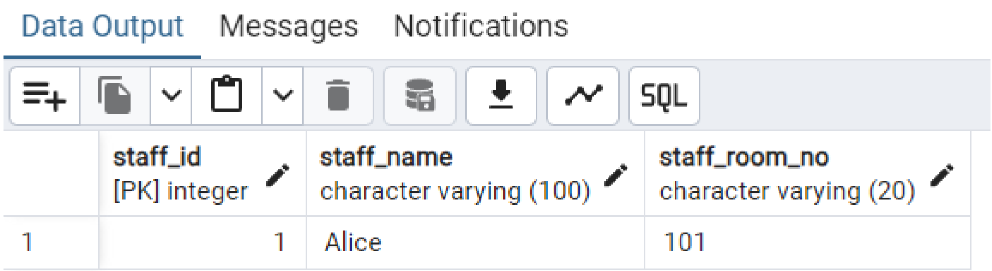
	printer_id [PK] integer	cost numeric (10,2)	increased_cost numeric
1	1	5.00	5.5000
2	2	7.50	8.2500
3	3	3.75	4.1250
4	4	6.00	6.6000

7. Search query using string operators

Use case: Find all staff whose name starts with 'A'.

Code:

```
SELECT *

FROM Staff

WHERE staff_name LIKE 'A%';
```



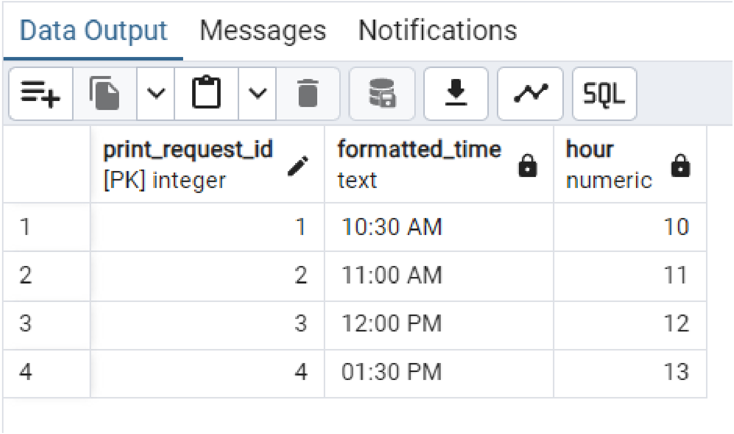
	staff_id [PK] integer	staff_name character varying (100)	staff_room_no character varying (20)
1	1	Alice	101

8. Usage of to_char, extract

Use case: Extract and format the time of all print requests.

Code:

```
SELECT print_request_id, TO_CHAR(time, 'HH:MI AM') AS formatted_time, EXTRACT(HOUR FROM time) AS hour

FROM Print_Request;
```



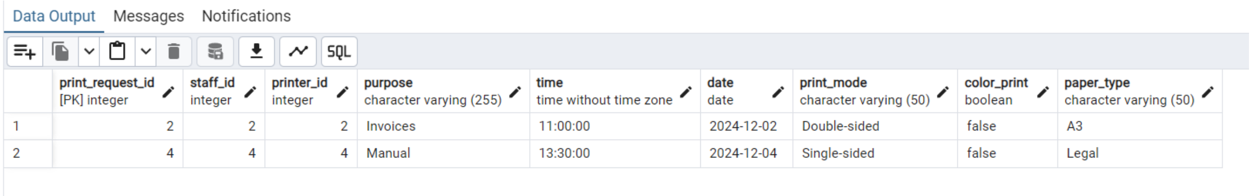
	print_request_id [PK] integer	formatted_time text	hour numeric
1	1	10:30 AM	10
2	2	11:00 AM	11
3	3	12:00 PM	12
4	4	01:30 PM	13

9. Between, IN, Not between, Not IN

Use case: Find all print requests made between '2024-12-02' and '2024-12-04', excluding requests made by staff 1 and 3.

Code:

```
SELECT *

FROM Print_Request

WHERE date BETWEEN '2024-12-02' AND '2024-12-04'

  AND staff_id NOT IN (1, 3);
```



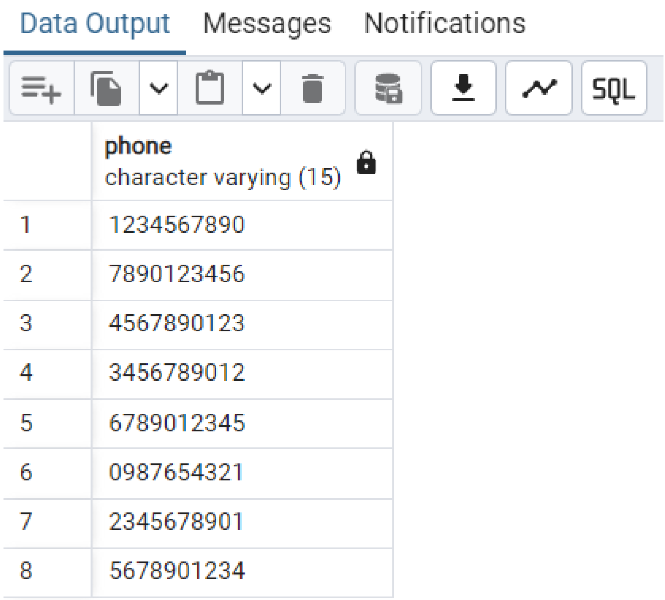
	print_request_id [PK] integer	staff_id integer	printer_id integer	purpose character varying (255)	time time without time zone	date date	print_mode character varying (50)	color_print boolean	paper_type character varying (50)
1	2	2	2	Invoices	11:00:00	2024-12-02	Double-sided	false	A3
2	4	4	4	Manual	13:30:00	2024-12-04	Single-sided	false	Legal

10. Set operations

Use case: Combine lists of phone numbers for both staff and delivery persons.

Code:

```
SELECT staff_phone AS phone

FROM Staff_Phone

UNION

SELECT delivery_person_phone AS phone

FROM Delivery_Phone;
```



	phone character varying (15)
1	1234567890
2	7890123456
3	4567890123
4	3456789012
5	6789012345
6	0987654321
7	2345678901
8	5678901234