CREATE TABLE with primary, foreign keys and NOT NULL

```sql
CREATE TABLE emloyers
    (id_boss SERIAL PRIMARY KEY,
    dep_Id INTEGER,
    boss_position CHARACTER VARYING(20) NOT NULL,
    firstName CHARACTER VARYING(30),
    lastName CHARACTER VARYING(30),
    age INTEGER,
    FOREIGN KEY (dep_Id) REFERENCES department (Id)
    );
```

ALTER and DROP

```sql
ALTER TABLE employers
ALTER COLUMN boss_position DROP NOT NULL;
```

Update wage rate to 6500 for employee with id 3

```sql
UPDATE employee
SET rate=6500
WHERE id_empl=3;
```

INSERT (completing the table)

```sql
INSERT INTO employers
(id_boss, dep_id, boss_position, firstname, lastname, age)
VALUES
(4, 3, 'boss assistant', 'Roman', 'Popel', 37),
(2, 4, 'assistant', 'Viktoria', 'Trush', 28),
(1, 1, 'assistant', 'Liubov', 'Manko', 30);
```

DELETE

```sql
DELETE FROM employers
WHERE boss_position='assistant';
```

DROP

```sql
DROP TABLE employers;
```

Show the list of product categories along with total ordering sums calculated for the orders made for the products of each category, during the year 1997

```sql
SELECT C.category_name,
ROUND(SUM(OD.quantity*(OD.unit_price-OD.unit_price*OD.discount)))
AS ordering_sums
FROM categories AS C
JOIN products AS P
ON P.category_id = C.category_id
JOIN order_details AS OD
ON OD.product_id = P.product_id
JOIN orders AS O
ON O.order_id = OD.order_id
WHERE O.order_date BETWEEN '1997-01-01' AND '1997-12-31'
GROUP BY category_name
ORDER BY ordering_sums ASC;
```

Show 3 oldest employees

```sql
SELECT first_name ||' '|| last_name AS employee,
EXTRACT (YEAR FROM AGE (CURRENT_DATE, birth_date)) AS ages_old
FROM employees
ORDER BY ages_old DESC
LIMIT 3;
```

Subquery

Show the list of customers' names who used to order the 'Chocolate'

```sql
SELECT customer_id
FROM customers
WHERE customer_id IN
    (SELECT customer_id
     FROM orders
     JOIN order_details ON orders.order_id = order_details.order_id
     JOIN products ON products.product_id = order_details.product_id
     WHERE product_name LIKE '%Chocolate%');
```