Database Project

Vîlceanu Diana-Maria

Contents

Description

I had always been passionate about arts and recently I turned my passion to graphic design. In the past few years, graphic design grew a lot and many companies hired designers to help them with their branding and image. Graphic design departments were created not so long ago and many niches have started to appear, such as UX/UI design.

Giving my passion about this field, I decided to construct a database based on a graphic design department schema, which is represented by the following tables:

The G_D_EMPLOYEES table, which shows us the information about the employees hired in the department, such as the employee id, the last name, first name and the salary.

The NICHES table, which contains information about the niches of graphic design an employee has chosen. Here, we find the niche id, the niche name and the grades.

The OFFICES table, which reffers to the locations of the offices our employees work in. The information consists of the office id, the office phone number and the locations.

The G_D_JOBS_HISTORY table, which shows information about the history the employees had. We find there the history id, the start date, the end date and the reason they left their previous job.

The G_D_ORDERS table, which reffers to the orders. It contains information such as order id, order date and the total value of the order.

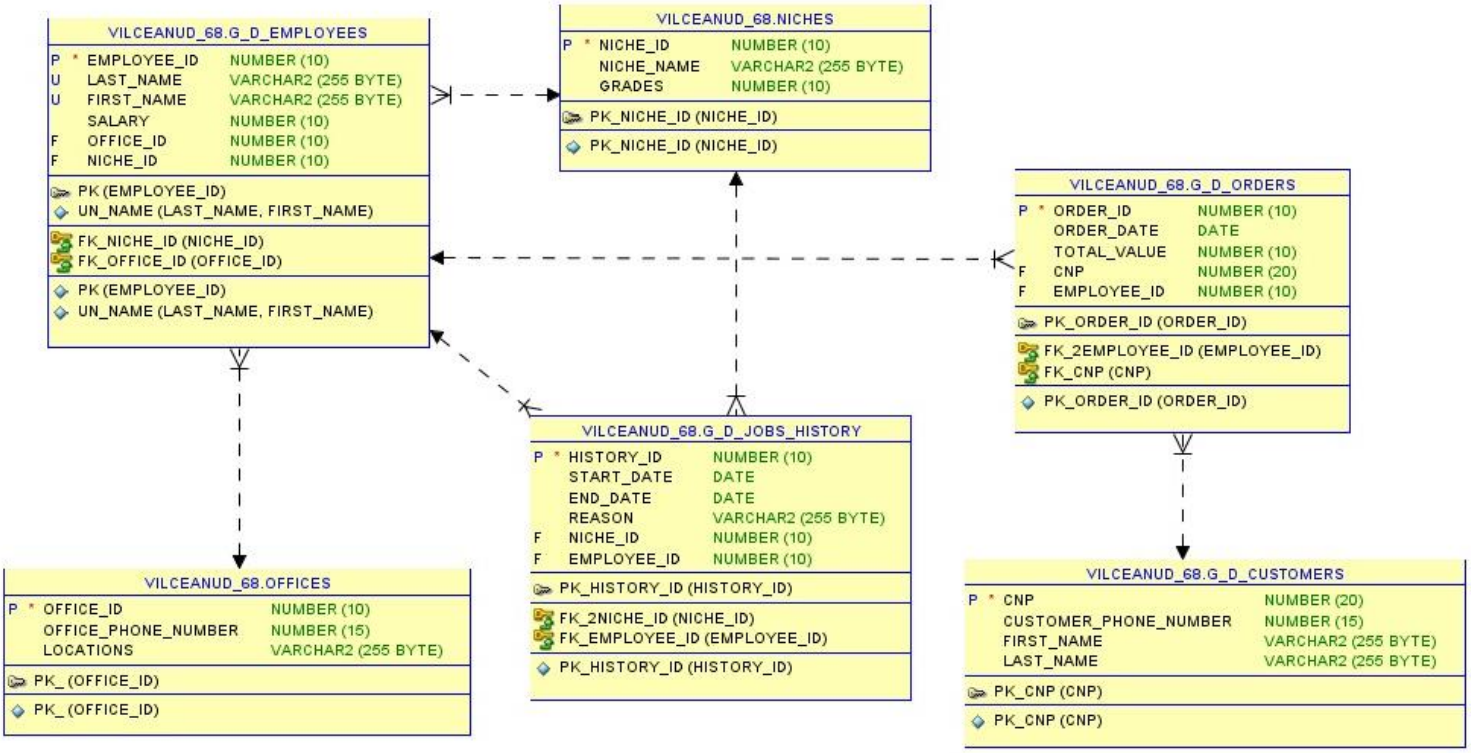
Last but not least, the G_D_CUSTOMERS table, which shows information about customers, such as customer CNP, customer phone number, first name and last name.

Links to the tables:

1. NICHES to G_D_EMPLOYEES one to many link – A niche contain many employees.
2. OFFICES to G_D_EMPLOYEES one to many link – An office contain many employees.
3. G_D_EMPLOYEES to G_D_JOBS_HISTORY one to many link – An employee can have many jobs in his/her history.
4. G_D_EMPLOYEES to G_D_ORDERS one to many link – An employee can deal with many orders.
5. NICHES to G_D_JOBS_HISTORY one to many link – A niche can have many jobs from history.
6. G_D_CUSTOMERS to G_D_ORDERS one to many link – A customer can place many orders.

On the next page, you can find the schema of the database I created:

Database schema



Constructing the database

At first, I created the tables without any constraint, because I decided to add them with the command ALTER TABLE to have a clear vision about linking my tables. Below there are the commands I used for this part:

```
CREATE TABLE G_D_EMPLOYEES(
EMPLOYEE_ID NUMBER(10),
LAST_NAME VARCHAR(255),
FIRST_NAME VARCHAR(255),
SALARY NUMBER(10)
);

CREATE TABLE NICHES(
NICHE_ID NUMBER(10),
NICHE_NAME VARCHAR(255),
GRADES NUMBER(10)
);

CREATE TABLE OFFICES(
OFFICE_ID NUMBER(10),
OFFICE_PHONE_NUMBER NUMBER(15),
LOCATIONS VARCHAR(255)
);

CREATE TABLE G_D_JOBS_HISTORY(
HISTORY_ID NUMBER(10),
START_DATE DATE,
END_DATE DATE,
REASON VARCHAR(255)
);

CREATE TABLE G_D_ORDERS(
ORDER_ID NUMBER(10),
```

ORDER_DATE DATE,

TOTAL_VALUE NUMBER(10)

);

CREATE TABLE G_D_CUSTOMERS(

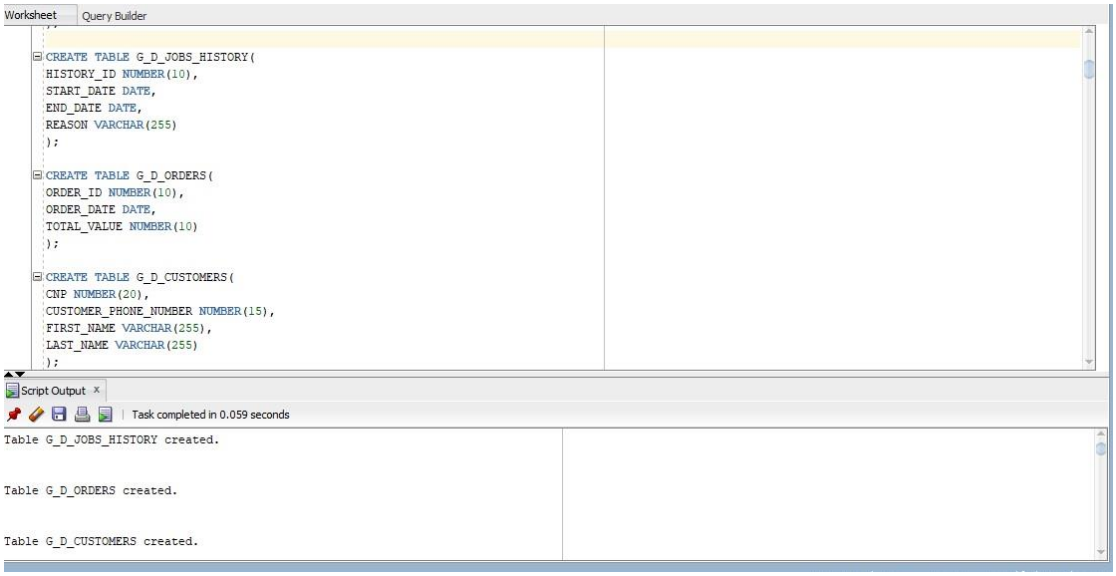
CNP NUMBER(20),

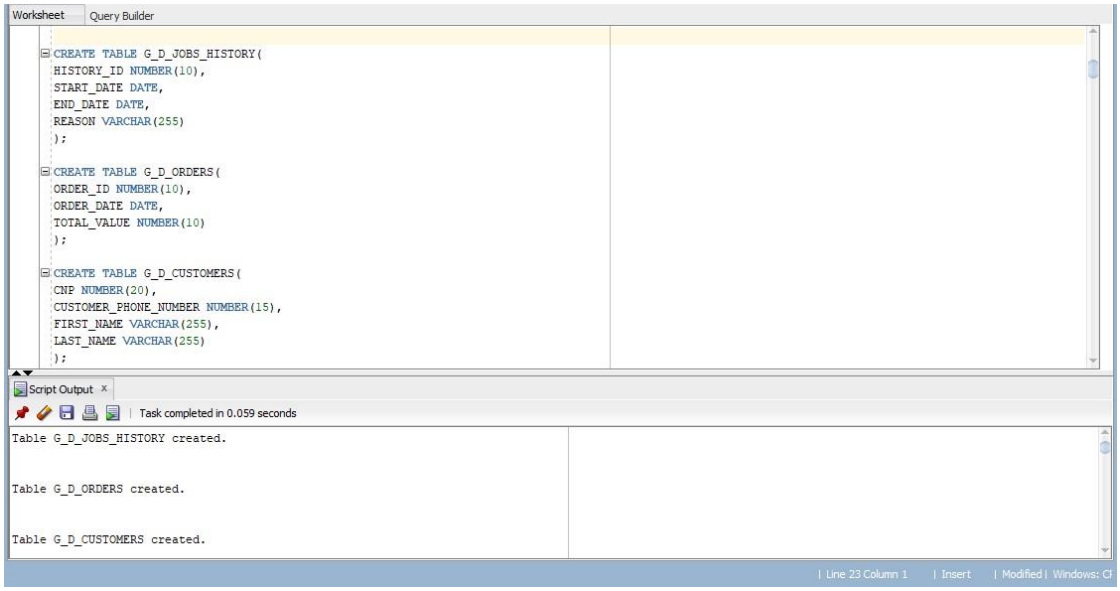
CUSTOMER_PHONE_NUMBER NUMBER(15),

FIRST_NAME VARCHAR(255),

LAST_NAME VARCHAR(255)

);

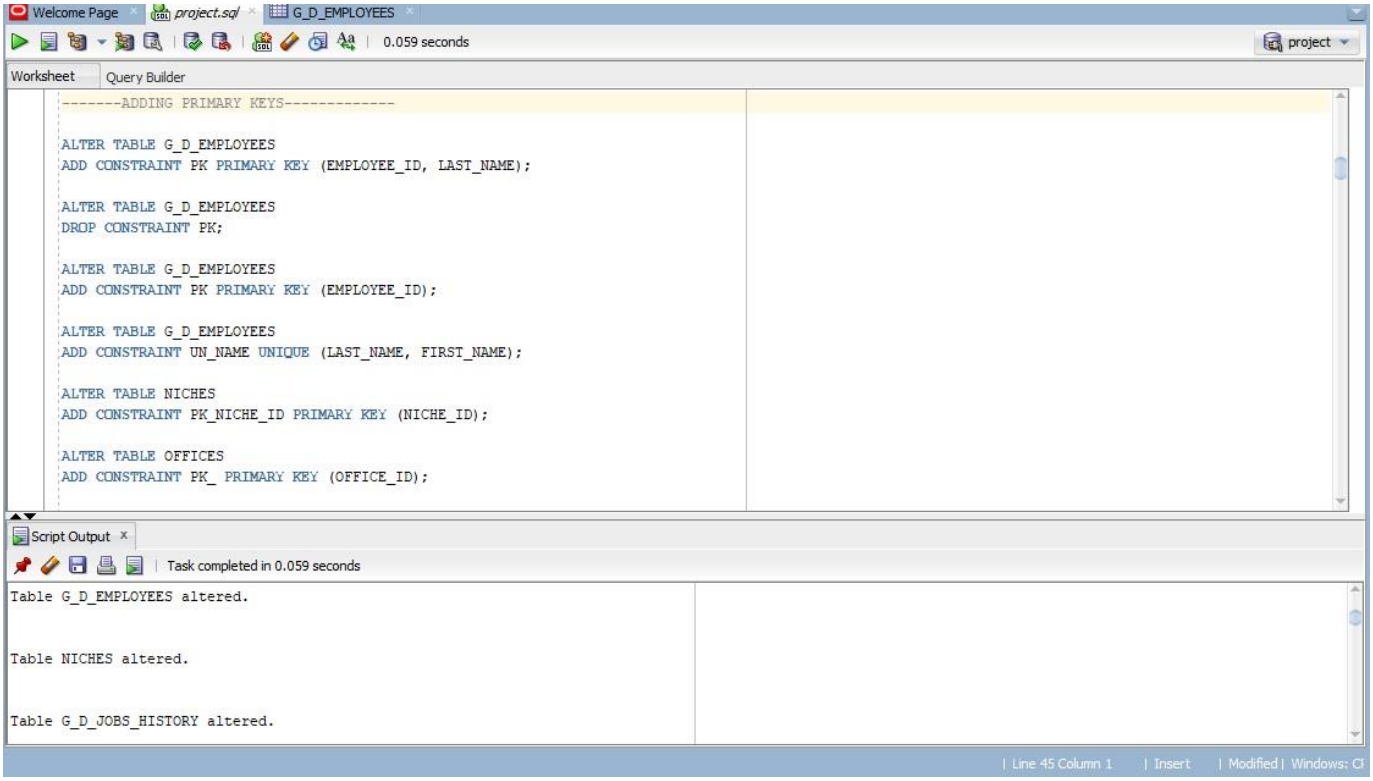




The second step for me was creating the primary keys for my tables. Here is how I did it:

```
ALTER TABLE G_D_EMPLOYEES
ADD CONSTRAINT PK PRIMARY KEY (EMPLOYEE_ID, LAST_NAME);

ALTER TABLE G_D_EMPLOYEES
DROP CONSTRAINT PK;

ALTER TABLE G_D_EMPLOYEES
ADD CONSTRAINT PK PRIMARY KEY (EMPLOYEE_ID);

ALTER TABLE G_D_EMPLOYEES
ADD CONSTRAINT UN_NAME UNIQUE (LAST_NAME, FIRST_NAME);

ALTER TABLE NICHES
ADD CONSTRAINT PK_NICHE_ID PRIMARY KEY (NICHE_ID);

ALTER TABLE OFFICES
ADD CONSTRAINT PK_ PRIMARY KEY (OFFICE_ID);

ALTER TABLE G_D_JOBS_HISTORY
ADD CONSTRAINT PK_HISTORY_ID PRIMARY KEY (HISTORY_ID);

ALTER TABLE G_D_ORDERS
ADD CONSTRAINT PK_ORDER_ID PRIMARY KEY (ORDER_ID);

ALTER TABLE G_D_CUSTOMERS
ADD CONSTRAINT PK_CNP PRIMARY KEY (CNP);
```

Note: For the primary key from the G_D_EMPLOYEES table I realised later that I need to change it to EMPLOYEE_ID in order to link the tables, so I altered it with the DROP CONSTRAINT command.



Then, I added the foreign keys:

ALTER TABLE G_D_EMPLOYEES

ADD OFFICE_ID NUMBER(10);

ALTER TABLE G_D_EMPLOYEES

ADD CONSTRAINT FK_OFFICE_ID FOREIGN KEY (OFFICE_ID) REFERENCES OFFICES(OFFICE_ID);

ALTER TABLE G_D_EMPLOYEES

ADD NICHE_ID NUMBER(10);

ALTER TABLE G_D_EMPLOYEES

ADD CONSTRAINT FK_NICHE_ID FOREIGN KEY (NICHE_ID) REFERENCES NICHES(NICHE_ID);

ALTER TABLE G_D_JOBS_HISTORY

ADD NICHE_ID NUMBER(10);

```
ALTER TABLE G_D_JOBS_HISTORY

ADD CONSTRAINT FK_2NICHE_ID FOREIGN KEY (NICHE_ID) REFERENCES
NICHES(NICHE_ID);


ALTER TABLE G_D_JOBS_HISTORY

ADD LAST_NAME VARCHAR(255);


ALTER TABLE G_D_JOBS_HISTORY

ADD FIRST_NAME VARCHAR(255);


ALTER TABLE G_D_JOBS_HISTORY

ADD CONSTRAINT FK_NAME_ID FOREIGN KEY (LAST_NAME, FIRST_NAME) REFERENCES
G_D_EMPLOYEES(LAST_NAME, FIRST_NAME);


ALTER TABLE G_D_ORDERS

ADD CNP NUMBER(20);


ALTER TABLE G_D_ORDERS

ADD LAST_NAME VARCHAR(255);


ALTER TABLE G_D_ORDERS

ADD FIRST_NAME VARCHAR(255);


ALTER TABLE G_D_ORDERS

ADD CONSTRAINT FK_2NAME_ID FOREIGN KEY (LAST_NAME, FIRST_NAME) REFERENCES
G_D_EMPLOYEES(LAST_NAME, FIRST_NAME);



ALTER TABLE G_D_ORDERS

ADD CONSTRAINT FK_CNP FOREIGN KEY (CNP) REFERENCES G_D_CUSTOMERS(CNP);
```

```
-------ADDING FOREIGN KEYS------

ALTER TABLE G_D_EMPLOYEES
ADD OFFICE_ID NUMBER(10);

ALTER TABLE G_D_EMPLOYEES
ADD CONSTRAINT FK_OFFICE_ID FOREIGN KEY (OFFICE_ID) REFERENCES OFFICES(OFFICE_ID);

ALTER TABLE G_D_EMPLOYEES
ADD NICHE_ID NUMBER(10);

ALTER TABLE G_D_EMPLOYEES
ADD CONSTRAINT FK_NICHE_ID FOREIGN KEY (NICHE_ID) REFERENCES NICHES(NICHE_ID);

ALTER TABLE G_D_JOBS_HISTORY
ADD NICHE_ID NUMBER(10);

ALTER TABLE G_D_JOBS_HISTORY
ADD CONSTRAINT FK_2NICHE_ID FOREIGN KEY (NICHE_ID) REFERENCES NICHES(NICHE_ID);
```

```
Table G_D_EMPLOYEES altered.

Table G_D_EMPLOYEES altered.

Table G_D_EMPLOYEES altered.
```

```
ALTER TABLE G_D_JOBS_HISTORY
ADD CONSTRAINT FK_2NICHE_ID FOREIGN KEY (NICHE_ID) REFERENCES NICHES(NICHE_ID);

ALTER TABLE G_D_JOBS_HISTORY
ADD LAST_NAME VARCHAR(255);

ALTER TABLE G_D_JOBS_HISTORY
ADD FIRST_NAME VARCHAR(255);

ALTER TABLE G_D_JOBS_HISTORY
ADD CONSTRAINT FK_NAME_ID FOREIGN KEY (LAST_NAME, FIRST_NAME) REFERENCES G_D_EMPLOYEES(LAST_NAME, FIRST_NAME);
```

Note: I made then some modifications to my tables because I realised I should keep EMPLOYEE_ID as a foreign key to make it easier to introduce data. Here they are:

```
ALTER TABLE G_D_JOBS_HISTORY

DROP CONSTRAINT FK_NAME_ID;

ALTER TABLE G_D_JOBS_HISTORY

DROP COLUMN FIRST_NAME;

ALTER TABLE G_D_JOBS_HISTORY

DROP COLUMN LAST_NAME;


ALTER TABLE G_D_ORDERS

DROP CONSTRAINT FK_2NAME_ID;
```

ALTER TABLE G_D_ORDERS

DROP COLUMN FIRST_NAME;

ALTER TABLE G_D_ORDERS

DROP COLUMN LAST_NAME;

ALTER TABLE G_D_JOBS_HISTORY

ADD EMPLOYEE_ID NUMBER(10);

ALTER TABLE G_D_ORDERS

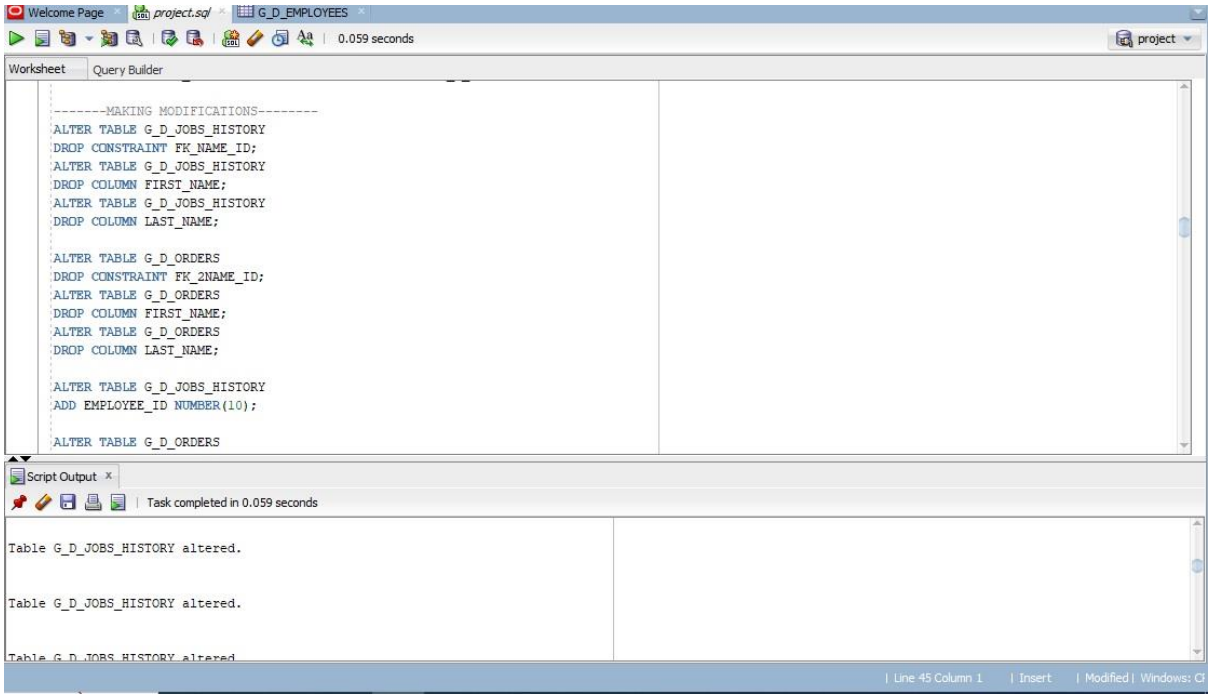
ADD EMPLOYEE_ID NUMBER(10);

ALTER TABLE G_D_JOBS_HISTORY

ADD CONSTRAINT FK_EMPLOYEE_ID FOREIGN KEY (EMPLOYEE_ID) REFERENCES G_D_EMPLOYEES(EMPLOYEE_ID);

ALTER TABLE G_D_ORDERS

ADD CONSTRAINT FK_2EMPLOYEE_ID FOREIGN KEY (EMPLOYEE_ID) REFERENCES G_D_EMPLOYEES(EMPLOYEE_ID);



Now that my tables were successfully created and altered, I moved on to the next step, inserting data.

Inserting data into tables

I started with the G_D_EMPLOYEES table. Below you can find how I introduced data in it:

```
INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (1, 'Smith', 'John', 50000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (2, 'Johnson', 'Jane', 55000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (3, 'Williams', 'Bob', 60000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (4, 'Jones', 'Sophie', 65000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (5, 'Brown', 'Michael', 70000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (6, 'Davis', 'Emma', 75000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (7, 'Miller', 'William', 80000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (8, 'Garcia', 'Emily', 85000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (9, 'Rodriguez', 'Jacob', 90000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
```

VALUES (10, 'Martinez', 'Isabella', 95000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (11, 'Anderson', 'Ethan', 100000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (12, 'Thomas', 'Ava', 105000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (13, 'Jackson', 'Madison', 110000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (14, 'White', 'Elizabeth', 115000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (15, 'Harris', 'Sofia', 120000);

```
-----EMPLOYEES---------

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (1, 'Smith', 'John', 50000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (2, 'Johnson', 'Jane', 55000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (3, 'Williams', 'Bob', 60000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (4, 'Jones', 'Sophie', 65000);

INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME, SALARY)
VALUES (5, 'Brown', 'Michael', 70000);
```

Here is how the tables look after inserting data:

	EMPLO...	LAST_NAME	FIRST_NA...	SALARY	OFFICE_ID	NICHE_ID
1	1	Smith	John	50000	(null)	(null)
2	2	Johnson	Jane	55000	(null)	(null)
3	3	Williams	Bob	60000	(null)	(null)
4	4	Jones	Sophie	65000	(null)	(null)
5	5	Brown	Michael	70000	(null)	(null)
6	6	Davis	Emma	75000	(null)	(null)
7	7	Miller	William	80000	(null)	(null)
8	8	Garcia	Emily	85000	(null)	(null)
9	9	Rodriguez	Jacob	90000	(null)	(null)
10	10	Martinez	Isabella	95000	(null)	(null)
11	11	Anderson	Ethan	100000	(null)	(null)
12	12	Thomas	Ava	105000	(null)	(null)
13	13	Jackson	Madison	110000	(null)	(null)
14	14	White	Elizabeth	115000	(null)	(null)
15	15	Harris	Sofia	120000	(null)	(null)

Then, I continued with the NICHES table. Below is the code I used:

```
INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (1, 'Branding', 9);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (2, 'Digital marketing', 8);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (3, 'Landing page web design', 7);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (4, 'UX/UI design', 8);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (5, 'Editorial design', 9);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (6, 'Illustration', 10);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (7, 'Type design', 8);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (8, '3D modelling', 7);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (9, 'Creative Arts', 8);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (10, 'Animation', 9);
```

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (11, 'Identity', 10);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (12, 'Product/Package Design', 7);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (13, 'Interfaces', 8);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (14, 'Corporate Designs', 9);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (15, 'Multimedia', 10);

```
-----------------NICHES-----------------------------------------------

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (1, 'Branding', 9);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (2, 'Digital marketing', 8);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (3, 'Landing page web design', 7);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (4, 'UX/UI design', 8);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (5, 'Editorial design', 9);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
VALUES (6, 'Illustration', 10);

INSERT INTO NICHES (NICHE_ID, NICHE_NAME, GRADES)
```

The table then looked like this:

	NICHE_ID	NICHE_NAME	GRADES
1	1	Branding	9
2	2	Digital marketing	8
3	3	Landing page web design	7
4	4	UX/UI design	8
5	5	Editorial design	9
6	6	Illustration	10
7	7	Type design	8
8	8	3D modelling	7
9	9	Creative Arts	8
10	10	Animation	9
11	11	Identity	10
12	12	Product/Package Design	7
13	13	Interfaces	8
14	14	Corporate Designs	9
15	15	Multimedia	10

The next table where I inserted data was the OFFICES table:

```
INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (1, '+1234567890', 'New York');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (2, '+0987654321', 'Los Angeles');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (3, '+1122334455', 'Chicago');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (4, '+6677889900', 'Houston');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (5, '+5544332211', 'Phoenix');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (6, '+4433221100', 'Philadelphia');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (7, '+3322110000', 'San Antonio');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (8, '+2211000000', 'San Diego');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (9, '+1100000000', 'Dallas');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (10, '+0000000000', 'San Jose');
```

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (11, '+1111111111', 'Austin');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (12, '+2222222222', 'Jacksonville');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (13, '+3333333333', 'Fort Worth');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (14, '+4444444444', 'Columbus');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (15, '+5555555555', 'San Francisco');

Worksheet | Query Builder

```
VALUES (15, 'Multimedia', 10);


--------------------OFFICES------------------------

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (1, '+1234567890', 'New York');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (2, '+0987654321', 'Los Angeles');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (3, '+1122334455', 'Chicago');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (4, '+6677889900', 'Houston');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
VALUES (5, '+5544332211', 'Phoenix');

INSERT INTO OFFICES (OFFICE_ID, OFFICE_PHONE_NUMBER, LOCATIONS)
```

The table then looked like this:

	OFFICE_ID	OFFICE_PHONE_NUMBER	LOCATIONS
1	1	1234567890	New York
2	2	987654321	Los Angeles
3	3	1122334455	Chicago
4	4	6677889900	Houston
5	5	5544332211	Phoenix
6	6	4433221100	Philadelphia
7	7	3322110000	San Antonio
8	8	2211000000	San Diego
9	9	1100000000	Dallas
10	10	0	San Jose
11	11	1111111111	Austin
12	12	2222222222	Jacksonville
13	13	3333333333	Fort Worth
14	14	4444444444	Columbus
15	15	5555555555	San Francisco

The next table I inserted data into was the G_D_JOBS_HISTORY table:

```
INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (1, DATE '2022-01-01', DATE '2022-12-30', 'Retired');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (2, DATE '2021-01-01', DATE '2021-12-31', 'Job completed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (3, DATE '2020-01-01', DATE '2020-12-31', 'Job changed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (4, DATE '2019-01-01', DATE '2019-12-31', 'Job completed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (5, DATE '2018-01-01', DATE '2018-12-31', 'Job changed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (6, DATE '2017-01-01', DATE '2017-12-31', 'Job completed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (7,DATE '2016-01-01',DATE '2016-12-31', 'Retired');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (8, DATE '2014-01-01', DATE '2014-12-31','Job changed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (9, DATE '2013-01-01', DATE '2013-12-31', 'Job completed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (10, DATE '2010-01-01',DATE '2010-12-31', 'Job completed');
```

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (11, DATE '2009-01-01', DATE '2009-12-31', 'Retired');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (12, DATE '2010-03-01', DATE '2020-12-31', 'Job changed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (13, DATE '2007-01-01', DATE '2007-12-31', 'Job completed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (14, DATE '2006-01-01', DATE '2006-12-31', 'Job changed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (15, DATE '2005-01-01', DATE '2005-12-31', 'Job completed');

```
----------------JOBS HISTORY----------------------------------

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (1, DATE '2022-01-01', DATE '2022-12-30', 'Retired');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (2, DATE '2021-01-01', DATE '2021-12-31', 'Job completed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (3, DATE '2020-01-01', DATE '2020-12-31', 'Job changed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (4, DATE '2019-01-01', DATE '2019-12-31', 'Job completed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (5, DATE '2018-01-01', DATE '2018-12-31', 'Job changed');

INSERT INTO G_D_JOBS_HISTORY (HISTORY_ID, START_DATE, END_DATE, REASON)
VALUES (6, DATE '2017-01-01', DATE '2017-12-31', 'Job completed');
```

The table after I inserted data in it, looks like this:

	HISTORY_ID	START_DATE	END_DATE	REASON	NICHE_ID	EMPLOYEE_ID
1	1	01-JAN-22	30-DEC-22	Retired	(null)	(null)
2	2	01-JAN-21	31-DEC-21	Job completed	(null)	(null)
3	3	01-JAN-20	31-DEC-20	Job changed	(null)	(null)
4	4	01-JAN-19	31-DEC-19	Job completed	(null)	(null)
5	5	01-JAN-18	31-DEC-18	Job changed	(null)	(null)
6	6	01-JAN-17	31-DEC-17	Job completed	(null)	(null)
7	7	01-JAN-16	31-DEC-16	Retired	(null)	(null)
8	8	01-JAN-14	31-DEC-14	Job changed	(null)	(null)
9	9	01-JAN-13	31-DEC-13	Job completed	(null)	(null)
10	10	01-JAN-10	31-DEC-10	Job completed	(null)	(null)
11	11	01-JAN-09	31-DEC-09	Retired	(null)	(null)
12	12	01-MAR-10	31-DEC-20	Job changed	(null)	(null)
13	13	01-JAN-07	31-DEC-07	Job completed	(null)	(null)
14	14	01-JAN-06	31-DEC-06	Job changed	(null)	(null)
15	15	01-JAN-05	31-DEC-05	Job completed	(null)	(null)

After that, I inserted data into the G_D_ORDERS table:

```
INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (1, DATE '2022-03-01', 100);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (2, DATE '2022-10-21', 240);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (3, DATE '2022-07-30', 73);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (4, DATE '2021-02-01', 560);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (5, DATE '2022-03-09', 760);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (6, DATE '2022-04-01', 100);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (7, DATE '2022-10-27', 267);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (8, DATE '2022-07-15', 89);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (9, DATE '2021-02-06', 97);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (10, DATE '2022-09-09', 547);
```

```
-------------------ORDERS-----------------------------------------------

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (1, DATE '2022-03-01', 100);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (2, DATE '2022-10-21', 240);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (3, DATE '2022-07-30', 73);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (4, DATE '2021-02-01', 560);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (5, DATE '2022-03-09', 760);

INSERT INTO G_D_ORDERS (ORDER_ID, ORDER_DATE, TOTAL_VALUE)
VALUES (6, DATE '2022-04-01', 100);
```

Script Output

Task completed in 0.154 seconds

```
1 row inserted.


1 row inserted.
```

The table looked like this:

	ORDER_ID	ORDER_DATE	TOTAL_VALUE	CNP	EMPLOYEE_ID
1	1	01-MAR-22	100	(null)	(null)
2	2	21-OCT-22	240	(null)	(null)
3	3	30-JUL-22	73	(null)	(null)
4	4	01-FEB-21	560	(null)	(null)
5	5	09-MAR-22	760	(null)	(null)
6	6	01-APR-22	100	(null)	(null)
7	7	27-OCT-22	267	(null)	(null)
8	8	15-JUL-22	89	(null)	(null)
9	9	06-FEB-21	97	(null)	(null)
10	10	09-SEP-22	547	(null)	(null)

At last, I inserted data into the G_D_CUSTOMERS table:

```
INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (6001012227376, 12025686179, 'John', 'Doe');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (5031227335939,15056441378, 'Jane', 'Smith');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (2910913267386, 12243112230, 'Bob', 'Johnson');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (6030925303231, 13209917679, 'Alice', 'Williams');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (6040525112942, 12035319681, 'Charlie', 'Jones');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (2890730422541, 15055004007, 'David', 'Brown');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (2890904014509,12195584968, 'Amy', 'Miller');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (1890912344377, 13808110551, 'James', 'Moore');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (2940104187581, 15055561044, 'Rachel', 'Taylor');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (2930107094845, 15056468477, 'Emily', 'Anderson');
```

```
----------CUSTOMERS------------------------------

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (6001012227376, 12025686179, 'John', 'Doe');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (5031227335939,15056441378, 'Jane', 'Smith');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (2910913267386, 12243112230, 'Bob', 'Johnson');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (6030925303231, 13209917679, 'Alice', 'Williams');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (6040525112942, 12035319681, 'Charlie', 'Jones');

INSERT INTO G_D_CUSTOMERS (CNP, CUSTOMER_PHONE_NUMBER, FIRST_NAME, LAST_NAME)
VALUES (2890730422541, 15055004007, 'David', 'Brown');
```

```
Task completed in 0.132 seconds

1 row inserted.

1 row inserted.
```

The table then looked like this:

	CNP	CUSTOMER_PHONE_NUMBER	FIRST_NAME	LAST_NAME
1	6001012227376	12025686179	John	Doe
2	5031227335939	15056441378	Jane	Smith
3	2910913267386	12243112230	Bob	Johnson
4	6030925303231	13209917679	Alice	Williams
5	6040525112942	12035319681	Charlie	Jones
6	2890730422541	15055004007	David	Brown
7	2890904014509	12195584968	Amy	Miller
8	1890912344377	13808110551	James	Moore
9	2940104187581	15055561044	Rachel	Taylor
10	2930107094845	15056468477	Emily	Anderson

I didn't forget to insert my name and my group in a table, as a part of the mandatory requirements. Here is the proof:

```
INSERT INTO G_D_EMPLOYEES (EMPLOYEE_ID, LAST_NAME, FIRST_NAME)
VALUES (1068, 'Vilceanu', 'Diana');
```

	EMPLO...	LAST_NAME	FIRST_NA...	SALARY	OFFICE_ID	NICHE_ID
1	1	Smith	John	50000	(null)	(null)
2	2	Johnson	Jane	55000	(null)	(null)
3	3	Williams	Bob	60000	(null)	(null)
4	4	Jones	Sophie	65000	(null)	(null)
5	5	Brown	Michael	70000	(null)	(null)
6	6	Davis	Emma	75000	(null)	(null)
7	7	Miller	William	80000	(null)	(null)
8	8	Garcia	Emily	85000	(null)	(null)
9	9	Rodriguez	Jacob	90000	(null)	(null)
10	10	Martinez	Isabella	95000	(null)	(null)
11	11	Anderson	Ethan	100000	(null)	(null)
12	12	Thomas	Ava	110000	(null)	(null)
13	13	Jackson	Madison	110000	(null)	(null)
14	14	White	Elizabeth	115000	(null)	(null)
15	15	Harris	Sofia	120000	(null)	(null)
16	1068	Vilceanu	Diana	(null)	(null)	(null)

The SELECT commands

1. Display the employees with the salary smaller than 70000.

SELECT * FROM G_D_EMPLOYEES

WHERE SALARY<70000;

```
SELECT * FROM G_D_EMPLOYEES
WHERE SALARY<70000;
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.015 seconds

	EMPLOYEE_ID	LAST_NAME	FIRST_NAME	SALARY	OFFICE_ID	NICHE_ID
1	1	Smith	John	50000	(null)	(null)
2	2	Johnson	Jane	55000	(null)	(null)
3	3	Williams	Bob	60000	(null)	(null)
4	4	Jones	Sophie	65000	(null)	(null)

2. Display the order with the order id 5.

SELECT * FROM G_D_ORDERS

WHERE ORDER_ID=5;

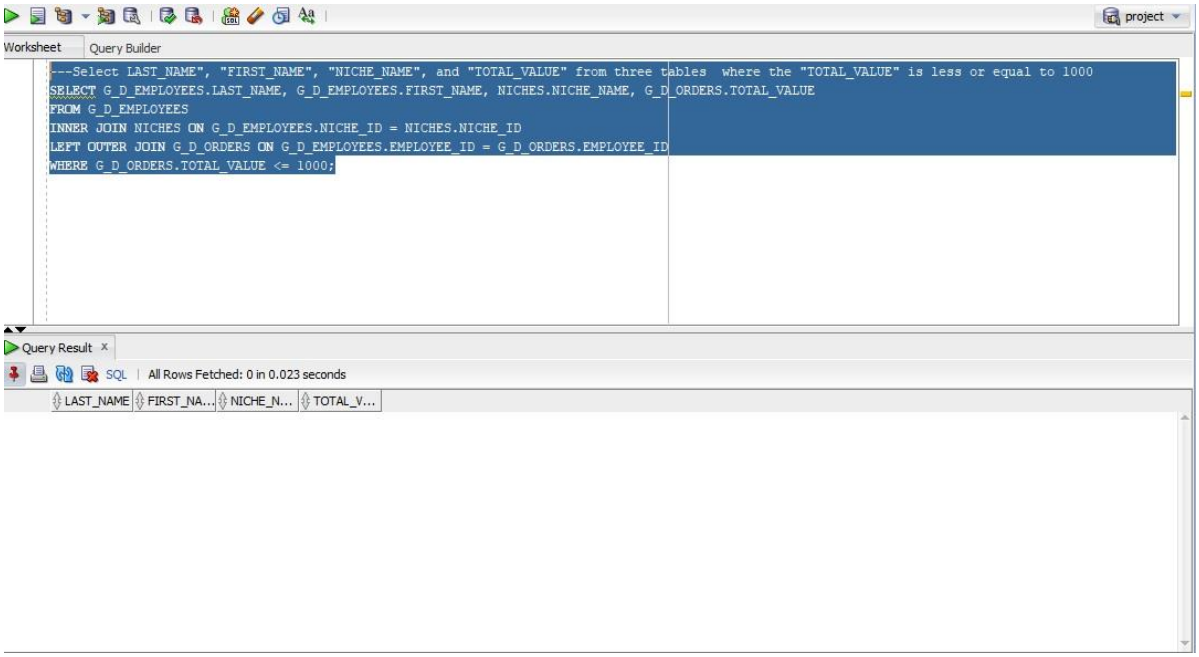
```
SELECT * FROM G_D_ORDERS
WHERE ORDER_ID=5;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.01 seconds

	ORDER_ID	ORDER_DATE	TOTAL_VALUE	CNP	EMPLOYEE_ID
1	5	09-MAR-22	760	(null)	(null)

3. Select LAST_NAME, FIRST_NAME, NICHE_NAME, and TOTAL_VALUE from three tables where the TOTAL_VALUE is less or equal to 1000



Note: For this command I had to update some rows from the G_D_EMPLOYEES table to match them with the NICHES table. Here is the code I used:

UPDATE G_D_EMPLOYEES

SET NICHE_ID=1

WHERE EMPLOYEE_ID=1;

UPDATE G_D_EMPLOYEES

SET NICHE_ID=2

WHERE EMPLOYEE_ID=2;

UPDATE G_D_EMPLOYEES

SET NICHE_ID=3

WHERE EMPLOYEE_ID=3;

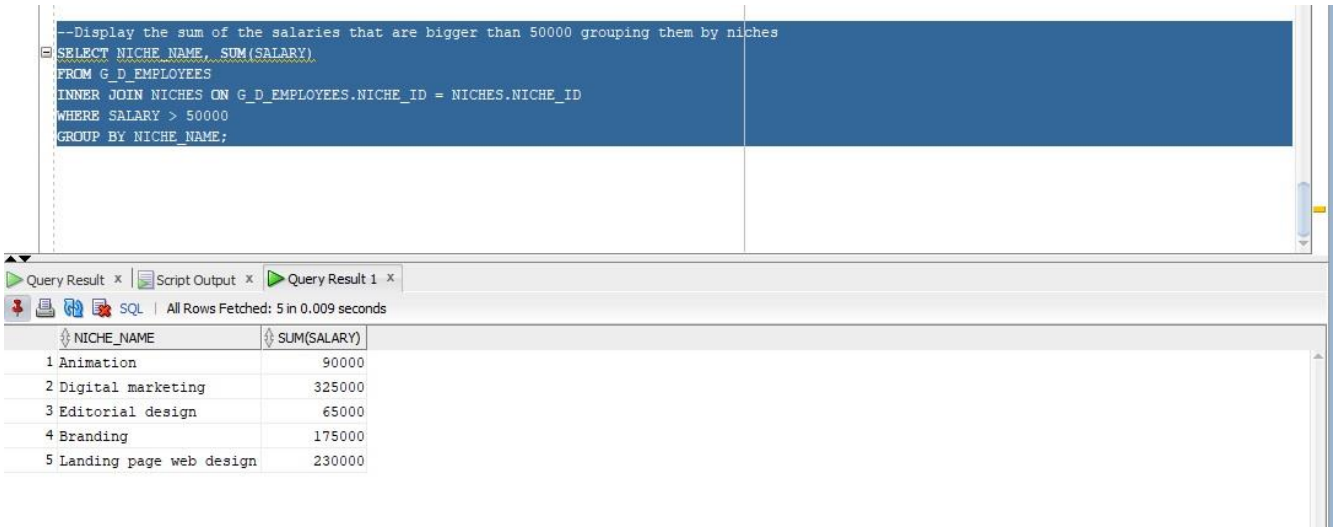
UPDATE G_D_EMPLOYEES

SET NICHE_ID=5

WHERE EMPLOYEE_ID=4;

```
UPDATE G_D_EMPLOYEES
SET NICHE_ID=3
WHERE EMPLOYEE_ID=5;

UPDATE G_D_EMPLOYEES
SET NICHE_ID=2
WHERE EMPLOYEE_ID=6;

UPDATE G_D_EMPLOYEES
SET NICHE_ID=1
WHERE EMPLOYEE_ID=7;

UPDATE G_D_EMPLOYEES
SET NICHE_ID=2
WHERE EMPLOYEE_ID=8;

UPDATE G_D_EMPLOYEES
SET NICHE_ID=10
WHERE EMPLOYEE_ID=9;

UPDATE G_D_EMPLOYEES
SET NICHE_ID=1
WHERE EMPLOYEE_ID=10;

UPDATE G_D_EMPLOYEES
SET NICHE_ID=3
WHERE EMPLOYEE_ID=11;

UPDATE G_D_EMPLOYEES
SET NICHE_ID=2
WHERE EMPLOYEE_ID=13;
```

4. Display the sum of the salaries that are bigger than 50000 grouping them by niches.
   SELECT NICHE_NAME, SUM(SALARY)
   FROM G_D_EMPLOYEES
   INNER JOIN NICHES ON G_D_EMPLOYEES.NICHE_ID = NICHES.NICHE_ID
   WHERE SALARY > 50000
   GROUP BY NICHE_NAME;



5. Display the order date, current time and order month of orders placed in 2022, in the specific format.

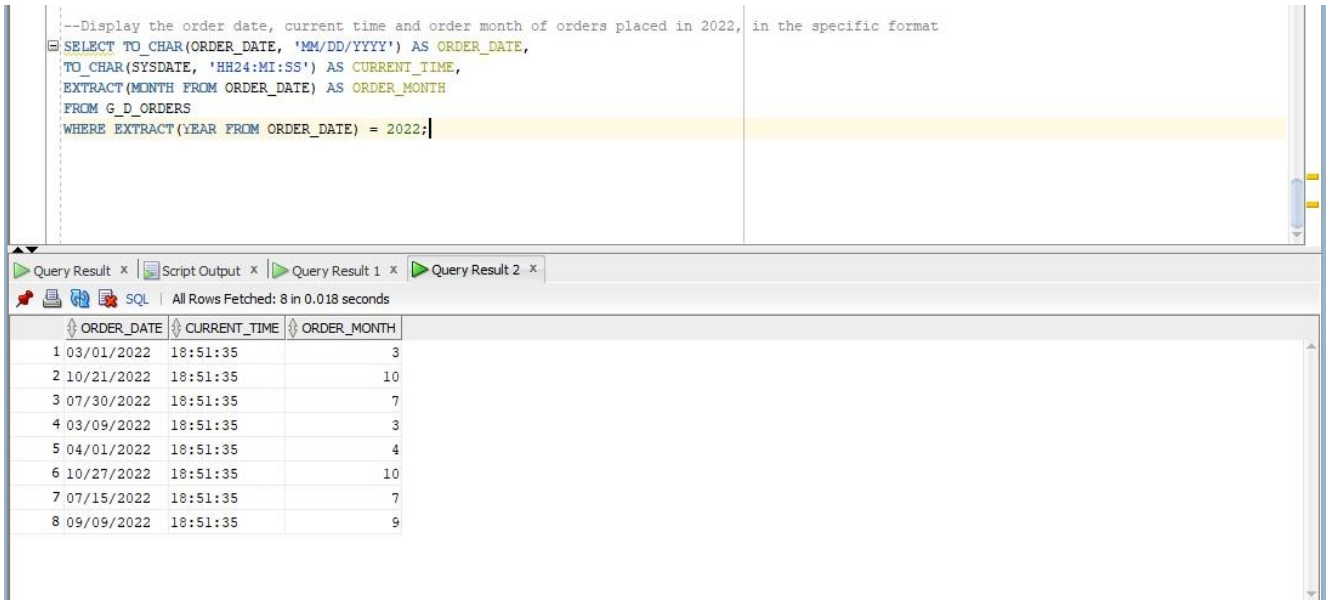
SELECT TO_CHAR(ORDER_DATE, 'MM/DD/YYYY') AS ORDER_DATE,

TO_CHAR(SYSDATE, 'HH24:MI:SS') AS CURRENT_TIME,

EXTRACT(MONTH FROM ORDER_DATE) AS ORDER_MONTH

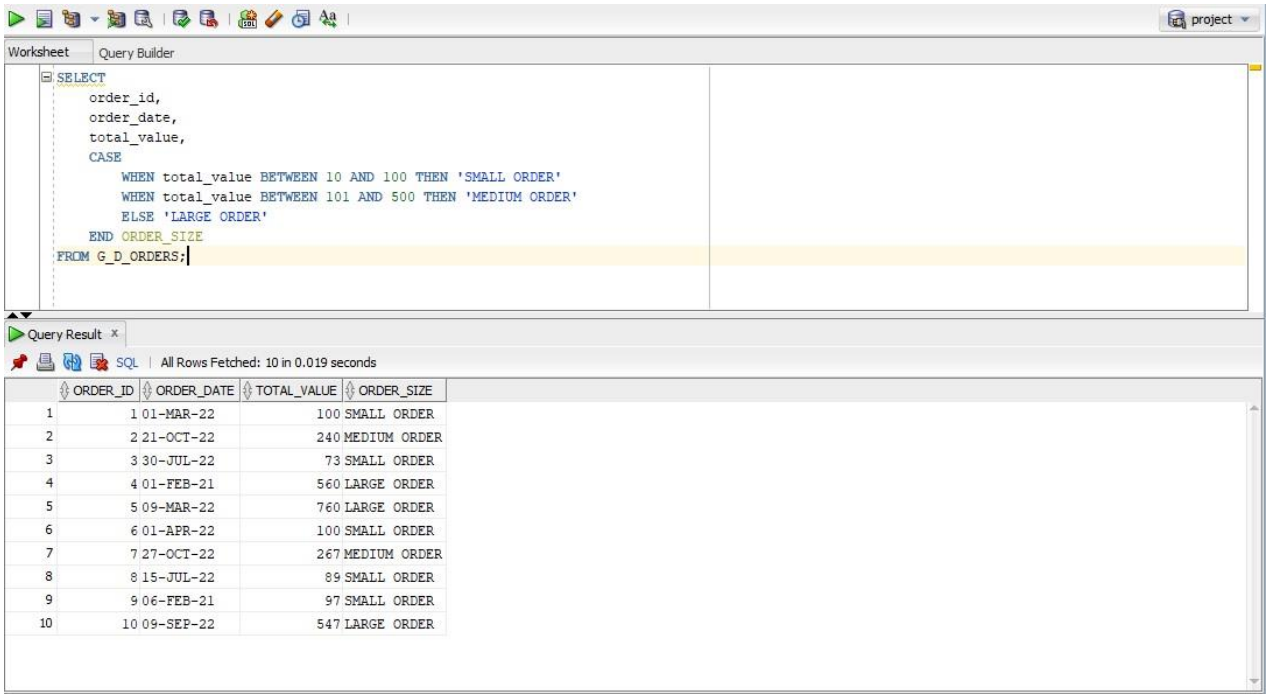
FROM G_D_ORDERS

WHERE EXTRACT(YEAR FROM ORDER_DATE) = 2022;



6. Display the order id, the order date and the total value of the orders by adding a new column which distributes each order by sizes.

```
SELECT
   order_id,
   order_date,
   total_value,
   CASE
      WHEN total_value BETWEEN 10 AND 100 THEN 'SMALL ORDER'
      WHEN total_value BETWEEN 101 AND 500 THEN 'MEDIUM ORDER'
      ELSE 'LARGE ORDER'
   END ORDER_SIZE
FROM G_D_ORDERS;
```



	ORDER_ID	ORDER_DATE	TOTAL_VALUE	ORDER_SIZE
1	1	01-MAR-22	100	SMALL ORDER
2	2	21-OCT-22	240	MEDIUM ORDER
3	3	30-JUL-22	73	SMALL ORDER
4	4	01-FEB-21	560	LARGE ORDER
5	5	09-MAR-22	760	LARGE ORDER
6	6	01-APR-22	100	SMALL ORDER
7	7	27-OCT-22	267	MEDIUM ORDER
8	8	15-JUL-22	89	SMALL ORDER
9	9	06-FEB-21	97	SMALL ORDER
10	10	09-SEP-22	547	LARGE ORDER

7. Display the employee types, knowing that a junior has a salary lower than 65000, a mid has a salary greater than 65000 and lower than 100000 and a senior has a salary greater than 100000.

```
SELECT
   employee_id,
   last_name,
   first_name,
   salary,
   decode(
   case
   when salary < 65000 then 1
```

when salary >= 65000 and salary <= 100000 then 2

when salary > 100000 then 3

end,

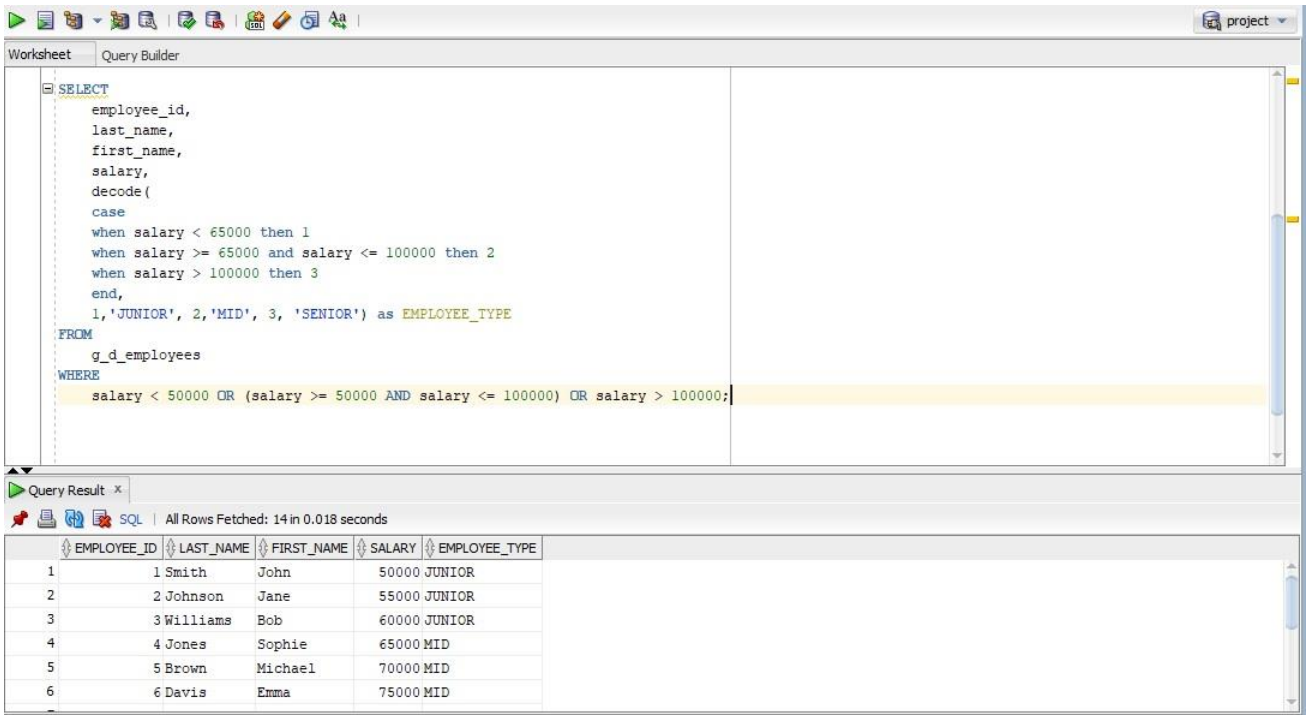
1,'JUNIOR', 2,'MID', 3, 'SENIOR') as EMPLOYEE_TYPE

FROM

g_d_employees

WHERE

salary < 50000 OR (salary >= 50000 AND salary <= 100000) OR salary > 100000;



8. Display the order id, order date and total value from G_D_ORDERS where total value hs any value and the date is bigger and lower than the January 1st 2022.

SELECT order_id, order_date, total_value

FROM G_D_ORDERS

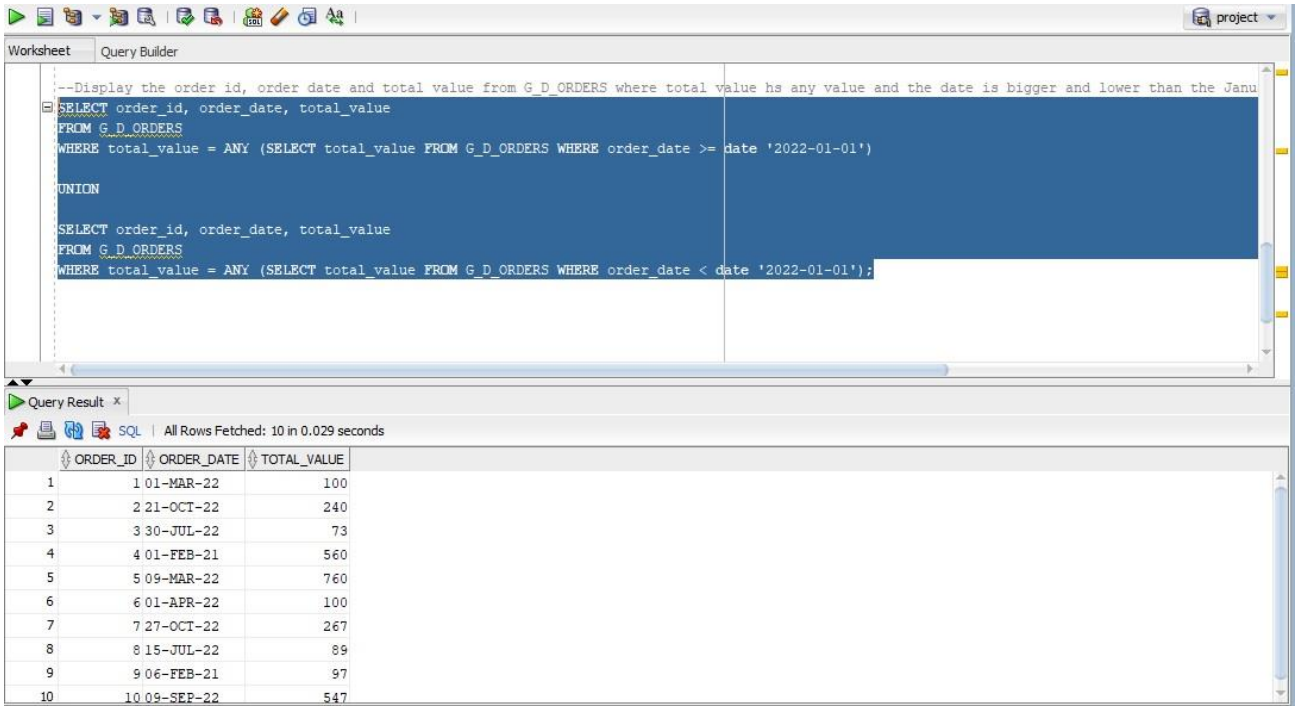
WHERE total_value = ANY (SELECT total_value FROM G_D_ORDERS WHERE order_date >= date '2022-01-01')

UNION

SELECT order_id, order_date, total_value

FROM G_D_ORDERS

WHERE total_value = ANY (SELECT total_value FROM G_D_ORDERS WHERE order_date < date '2022-01-01');



```
--Display the order id, order date and total value from G_D_ORDERS where total value hs any value and the date is bigger and lower than the Janu
SELECT order_id, order_date, total_value
FROM G_D_ORDERS
WHERE total_value = ANY (SELECT total_value FROM G_D_ORDERS WHERE order_date >= date '2022-01-01')

UNION

SELECT order_id, order_date, total_value
FROM G_D_ORDERS
WHERE total_value = ANY (SELECT total_value FROM G_D_ORDERS WHERE order_date < date '2022-01-01');
```

All Rows Fetched: 10 in 0.029 seconds

	ORDER_ID	ORDER_DATE	TOTAL_VALUE
1	1	01-MAR-22	100
2	2	21-OCT-22	240
3	3	30-JUL-22	73
4	4	01-FEB-21	560
5	5	09-MAR-22	760
6	6	01-APR-22	100
7	7	27-OCT-22	267
8	8	15-JUL-22	89
9	9	06-FEB-21	97
10	10	09-SEP-22	547

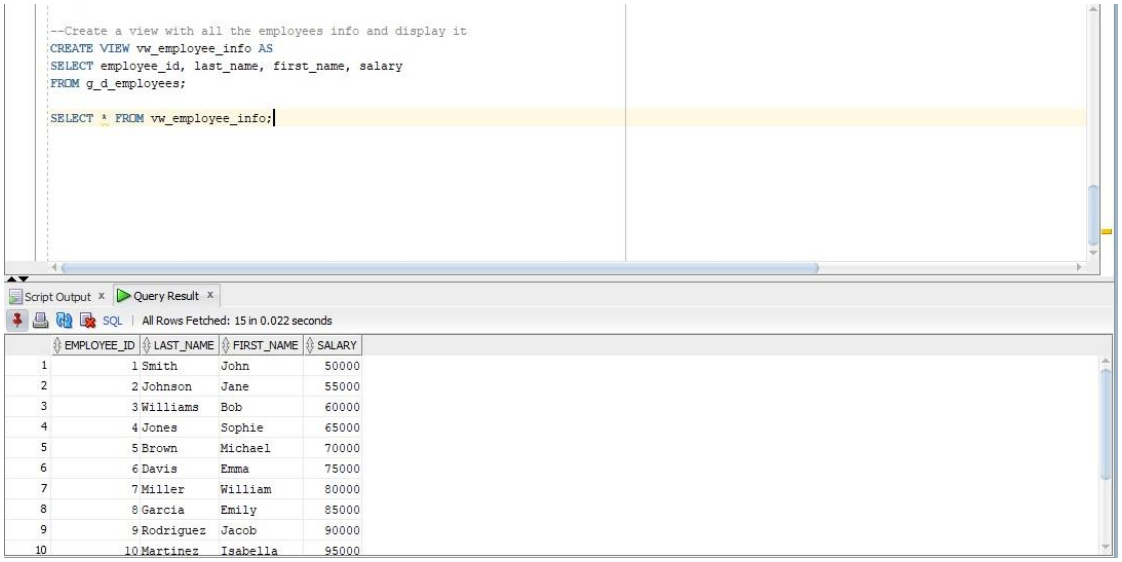
9. Create a view with all the employees info and display it.

CREATE VIEW vw_employee_info AS

SELECT employee_id, last_name, first_name, salary

FROM g_d_employees;

SELECT * FROM vw_employee_info;



```
--Create a view with all the employees info and display it
CREATE VIEW vw_employee_info AS
SELECT employee_id, last_name, first_name, salary
FROM g_d_employees;

SELECT * FROM vw_employee_info;
```

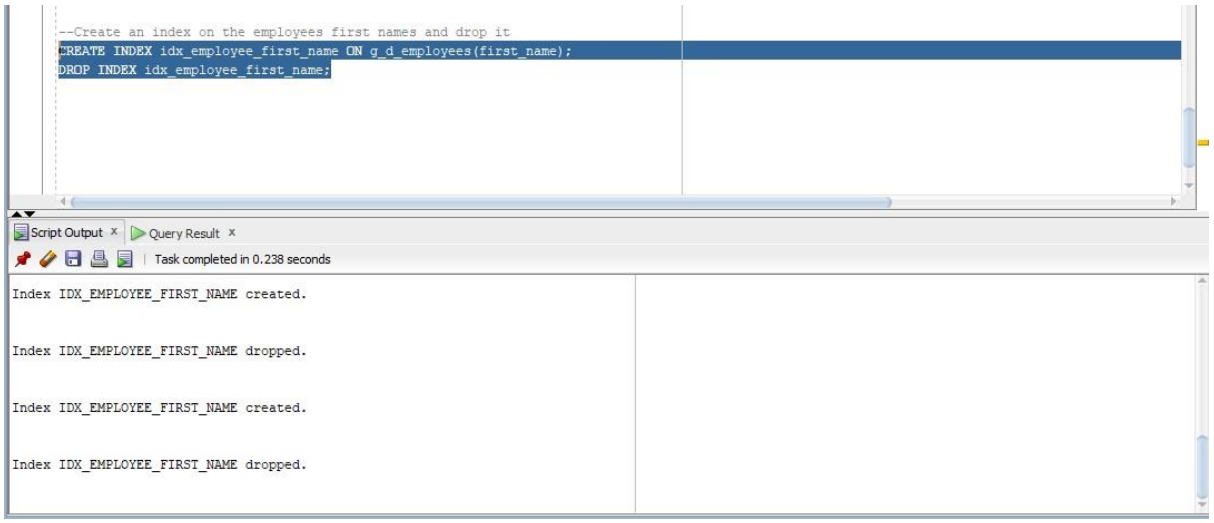
All Rows Fetched: 15 in 0.022 seconds

	EMPLOYEE_ID	LAST_NAME	FIRST_NAME	SALARY
1	1	Smith	John	50000
2	2	Johnson	Jane	55000
3	3	Williams	Bob	60000
4	4	Jones	Sophie	65000
5	5	Brown	Michael	70000
6	6	Davis	Emma	75000
7	7	Miller	William	80000
8	8	Garcia	Emily	85000
9	9	Rodriguez	Jacob	90000
10	10	Martinez	Isabella	95000

10. Create an index on the employees first names and drop it.

CREATE INDEX idx_employee_first_name ON g_d_employees(first_name);

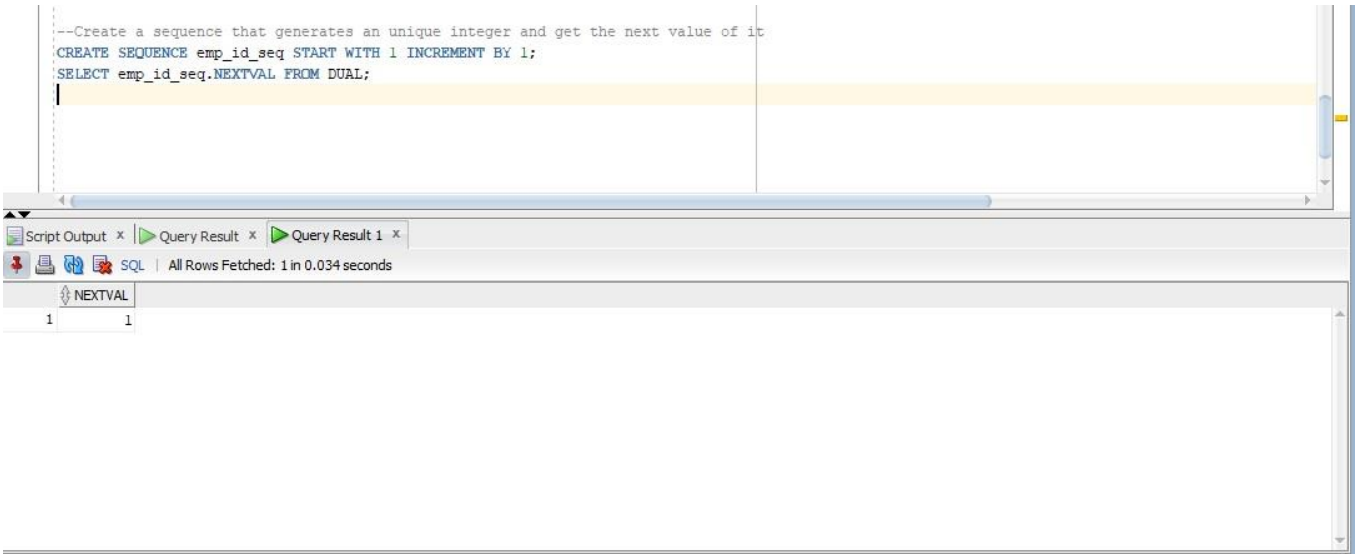
DROP INDEX idx_employee_first_name;



11. Create a sequence that generates an unique integer and get the next value of it.

CREATE SEQUENCE emp_id_seq START WITH 1 INCREMENT BY 1;

SELECT emp_id_seq.NEXTVAL FROM DUAL;



12. Display the history id, the start and end date of the employees that has changed their job.

SELECT

history_id,

to_char(start_date, 'MM/DD/YYYY') as start_date,

to_char(end_date, 'MM/DD/YYYY') as end_date,

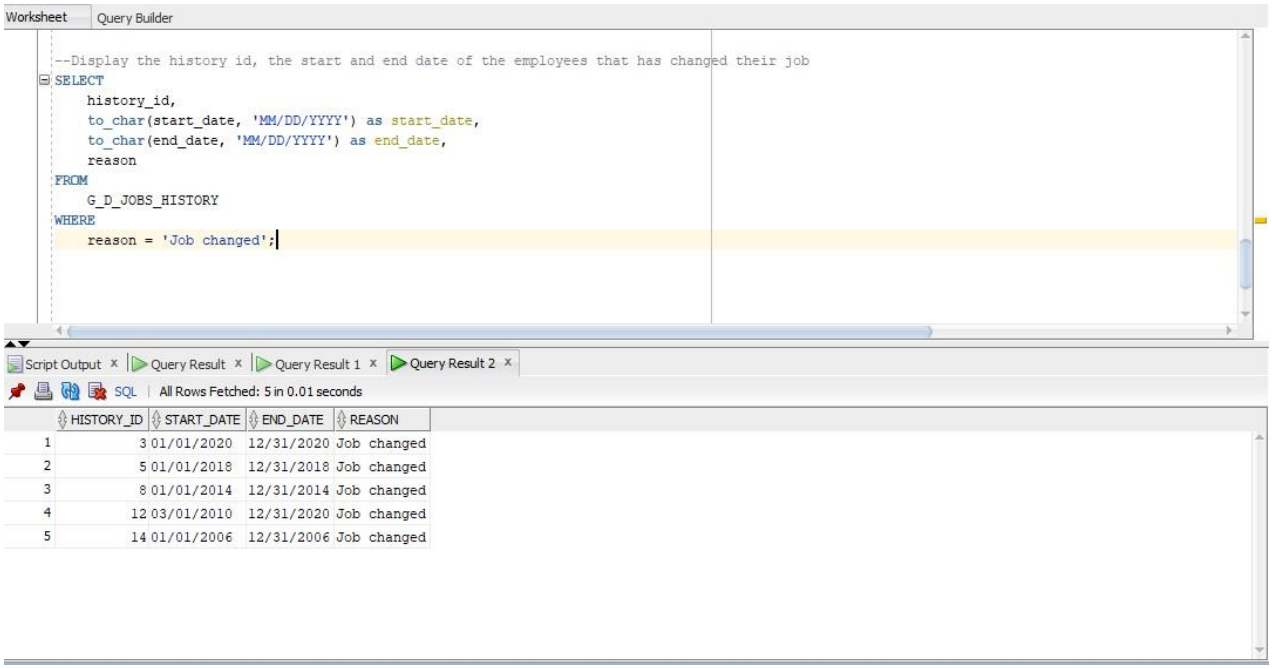
reason

FROM

G_D_JOBS_HISTORY

WHERE

reason = 'Job changed';



13. Display the history_id, start date, end date and reason for the employees that ended their job before the system date.

SELECT

history_id,

start_date,

end_date,

reason

FROM

G_D_JOBS_HISTORY

WHERE

end_date <= sysdate;

Worksheet | Query Builder

```
--Display the history_id, start date, end date and reason for the employees that ended their job before the system date
SELECT
    history_id,
    start_date,
    end_date,
    reason
FROM
    G_D_JOBS_HISTORY
WHERE
    end_date <= sysdate;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 15 in 0.013 seconds

	HISTORY_ID	START_DATE	END_DATE	REASON
1	1	01-JAN-22	30-DEC-22	Retired
2	2	01-JAN-21	31-DEC-21	Job completed
3	3	01-JAN-20	31-DEC-20	Job changed
4	4	01-JAN-19	31-DEC-19	Job completed
5	5	01-JAN-18	31-DEC-18	Job changed
6	6	01-JAN-17	31-DEC-17	Job completed
7	7	01-JAN-16	31-DEC-16	Retired
8	8	01-JAN-14	31-DEC-14	Job changed
9	9	01-JAN-13	31-DEC-13	Job completed
10	10	01-JAN-10	31-DEC-10	Job completed

14. Display the id of the employees that started their previous job in January and have the niche id 2.

```
SELECT
   HISTORY_ID
FROM
   G_D_JOBS_HISTORY
WHERE
   EXTRACT(MONTH FROM start_date) = 1
INTERSECT
SELECT
  EMPLOYEE_ID
FROM
   G_D_EMPLOYEES
WHERE
   NICHE_ID= 2;
```

Worksheet | Query Builder

```
--Display the id of the employees that started their previous job in January and have the niche id 2
SELECT
    HISTORY_ID
FROM
    G_D_JOBS_HISTORY
WHERE
    EXTRACT(MONTH FROM start_date) = 1
INTERSECT
SELECT
   EMPLOYEE_ID
FROM
    G_D_EMPLOYEES
WHERE
    NICHE_ID= 2;
```

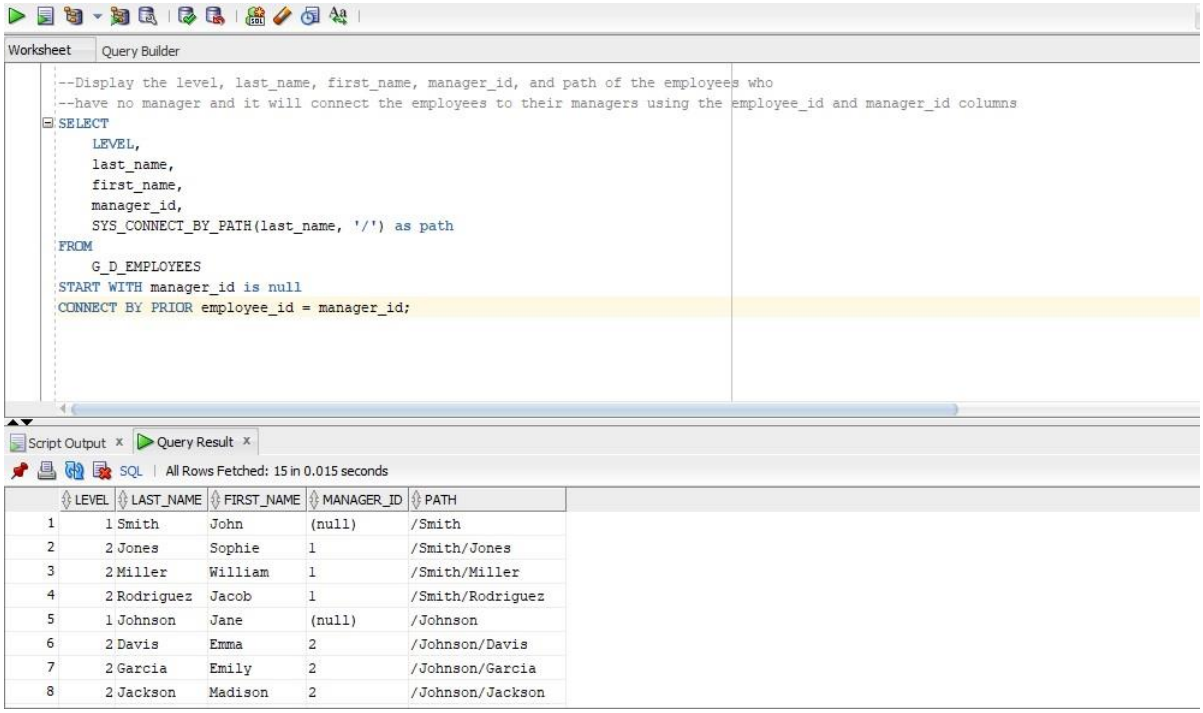
Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.013 seconds

	HISTORY_ID
1	2
2	6
3	8
4	13

15. Display the level, last_name, first_name, manager_id, and path of the employees who have no manager and it will connect the employees to their managers using the employee_id and manager_id columns.

```
SELECT
    LEVEL,
    last_name,
    first_name,
    manager_id,
    SYS_CONNECT_BY_PATH(last_name, '/') as path
FROM
    G_D_EMPLOYEES
START WITH manager_id is null
CONNECT BY PRIOR employee_id = manager_id;
```



	LEVEL	LAST_NAME	FIRST_NAME	MANAGER_ID	PATH
1	1	Smith	John	(null)	/Smith
2	2	Jones	Sophie	1	/Smith/Jones
3	2	Miller	William	1	/Smith/Miller
4	2	Rodriguez	Jacob	1	/Smith/Rodriguez
5	1	Johnson	Jane	(null)	/Johnson
6	2	Davis	Emma	2	/Johnson/Davis
7	2	Garcia	Emily	2	/Johnson/Garcia
8	2	Jackson	Madison	2	/Johnson/Jackson