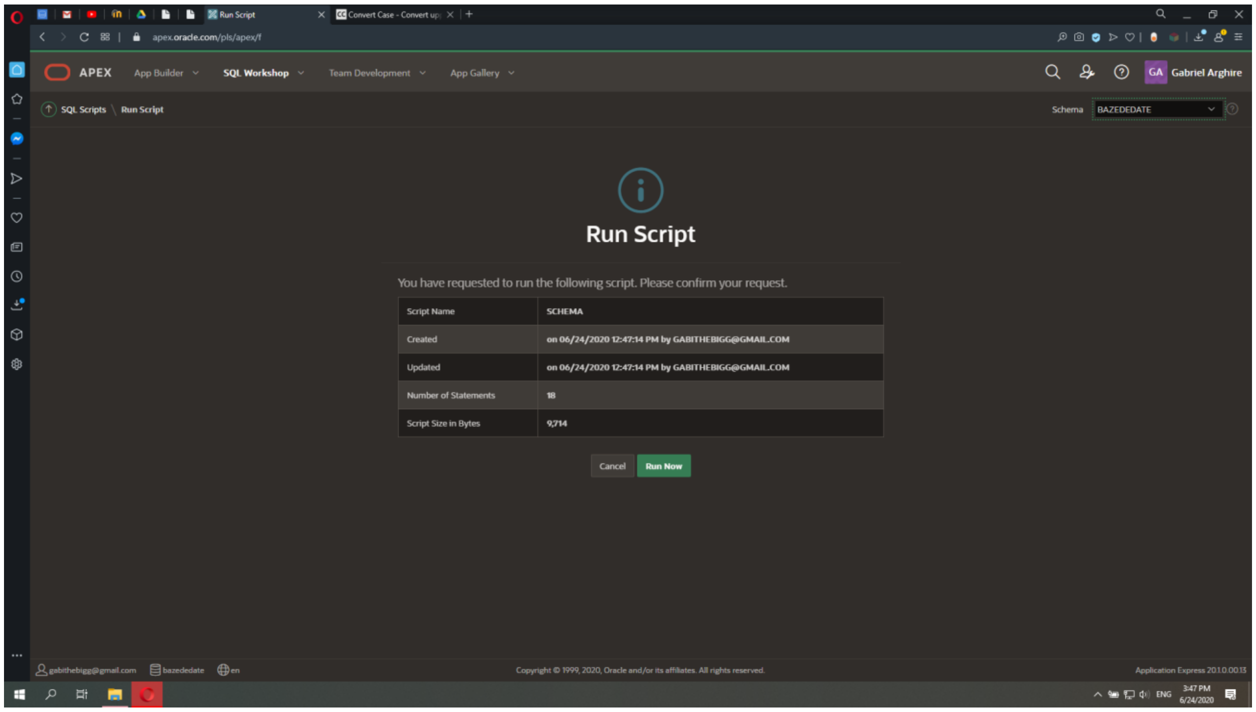
Queries output

Using Apex Oracle

Running script called ”SCHEMA” for creation of tables, adding constraints and inserting data.



Run Script

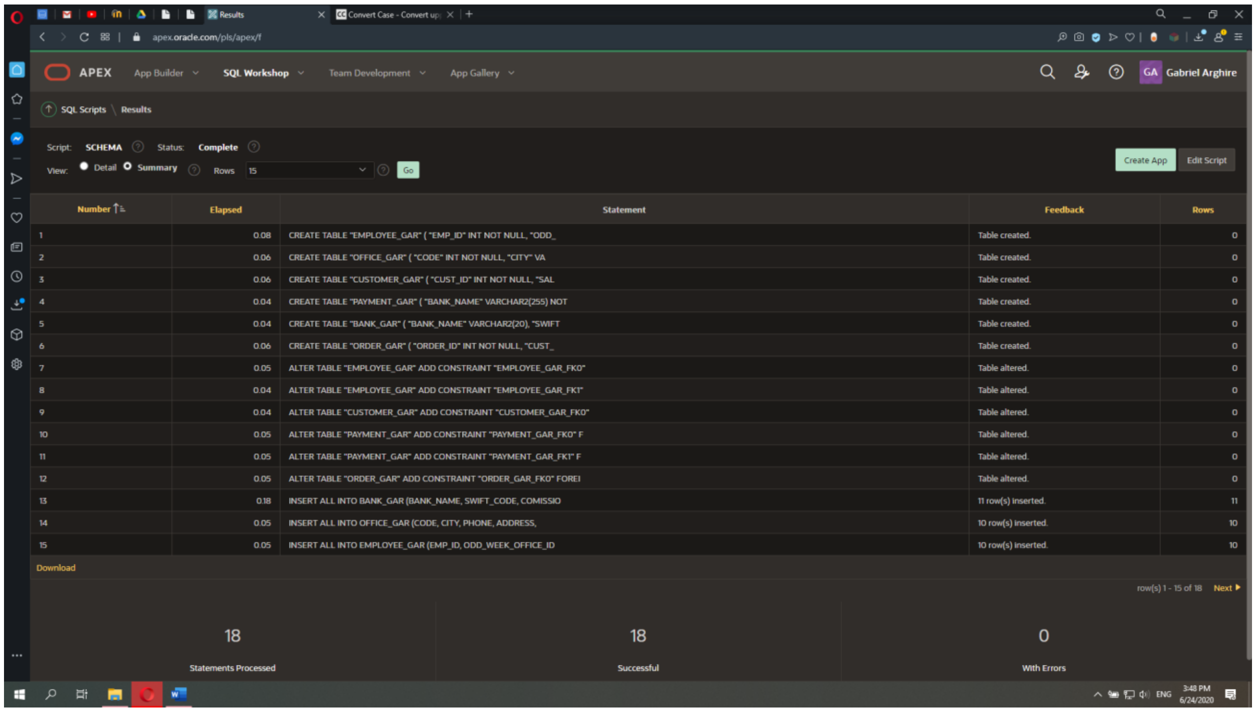
You have requested to run the following script. Please confirm your request.

Script Name	SCHEMA
Created	on 06/24/2020 12:47:14 PM by GABITHEBIGG@GMAIL.COM
Updated	on 06/24/2020 12:47:14 PM by GABITHEBIGG@GMAIL.COM
Number of Statements	18
Script Size in Bytes	9,714

Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.1.0.00.13

Results from page 1, after running script ”SCHEMA”



Script: SCHEMA Status: Complete
View: Detail [✓ Summary] Rows 15

Number	Elapsed	Statement	Feedback	Rows
1	0.08	CREATE TABLE "EMPLOYEE_GAR" ("EMP_ID" INT NOT NULL, "ODD_	Table created.	0
2	0.06	CREATE TABLE "OFFICE_GAR" ("CODE" INT NOT NULL, "CITY" VA	Table created.	0
3	0.06	CREATE TABLE "CUSTOMER_GAR" ("CUST_ID" INT NOT NULL, "SAL	Table created.	0
4	0.04	CREATE TABLE "PAYMENT_GAR" ("BANK_NAME" VARCHAR2(255) NOT	Table created.	0
5	0.04	CREATE TABLE "BANK_GAR" ("BANK_NAME" VARCHAR2(20), "SWIFT	Table created.	0
6	0.06	CREATE TABLE "ORDER_GAR" ("ORDER_ID" INT NOT NULL, "CUST_	Table created.	0
7	0.05	ALTER TABLE "EMPLOYEE_GAR" ADD CONSTRAINT "EMPLOYEE_GAR_FK0"	Table altered.	0
8	0.04	ALTER TABLE "EMPLOYEE_GAR" ADD CONSTRAINT "EMPLOYEE_GAR_FK1"	Table altered.	0
9	0.04	ALTER TABLE "CUSTOMER_GAR" ADD CONSTRAINT "CUSTOMER_GAR_FK0"	Table altered.	0
10	0.05	ALTER TABLE "PAYMENT_GAR" ADD CONSTRAINT "PAYMENT_GAR_FK0" F	Table altered.	0
11	0.05	ALTER TABLE "PAYMENT_GAR" ADD CONSTRAINT "PAYMENT_GAR_FK1" F	Table altered.	0
12	0.05	ALTER TABLE "ORDER_GAR" ADD CONSTRAINT "ORDER_GAR_FK0" FOREI	Table altered.	0
13	0.18	INSERT ALL INTO BANK_GAR (BANK_NAME, SWIFT_CODE, COMISSIO	11 row(s) inserted.	11
14	0.05	INSERT ALL INTO OFFICE_GAR (CODE, CITY, PHONE, ADDRESS,	10 row(s) inserted.	10
15	0.05	INSERT ALL INTO EMPLOYEE_GAR (EMP_ID, ODD_WEEK_OFFICE_ID	10 row(s) inserted.	10

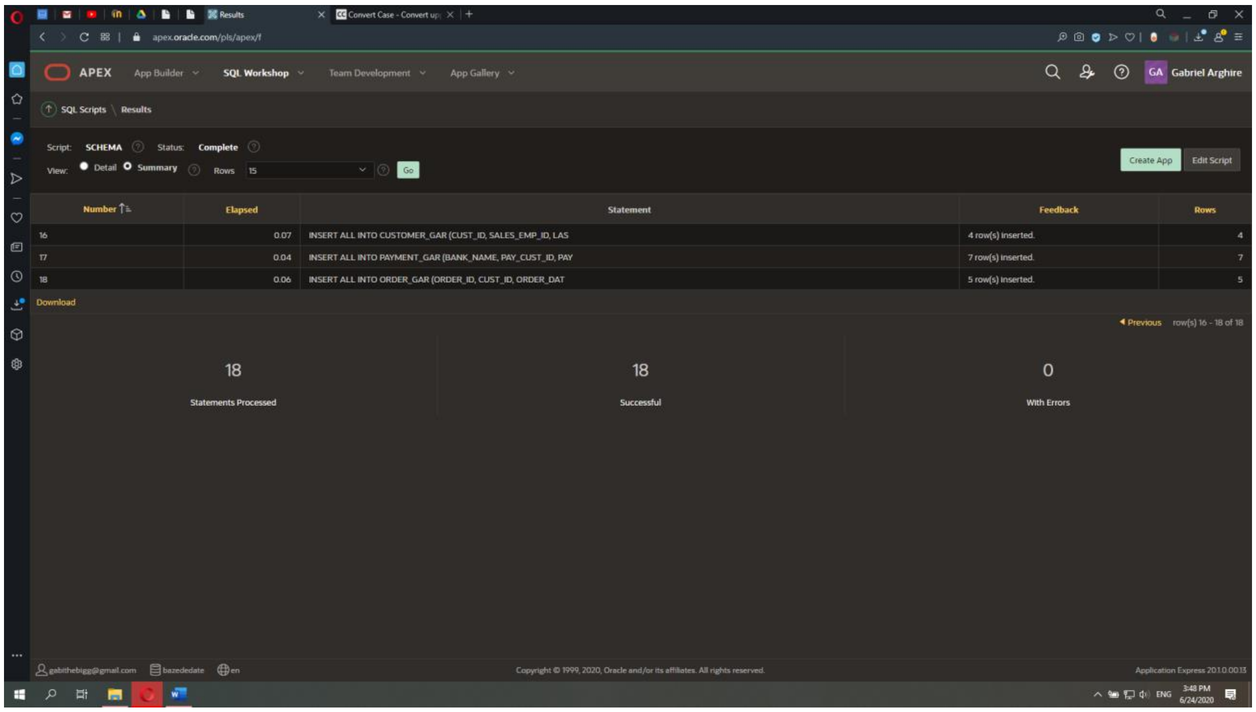
row(s) 1 - 15 of 18

18 Statements Processed

18 Successful

0 With Errors

Results from page 2, after running script ”SCHEMA”



Script: SCHEMA Status: Complete
View: Detail [✓ Summary] Rows 15

Number	Elapsed	Statement	Feedback	Rows
16	0.07	INSERT ALL INTO CUSTOMER_GAR (CUST_ID, SALES_EMP_ID, LAS	4 row(s) inserted.	4
17	0.04	INSERT ALL INTO PAYMENT_GAR (BANK_NAME, PAY_CUST_ID, PAY	7 row(s) inserted.	7
18	0.06	INSERT ALL INTO ORDER_GAR (ORDER_ID, CUST_ID, ORDER_DAT	5 row(s) inserted.	5

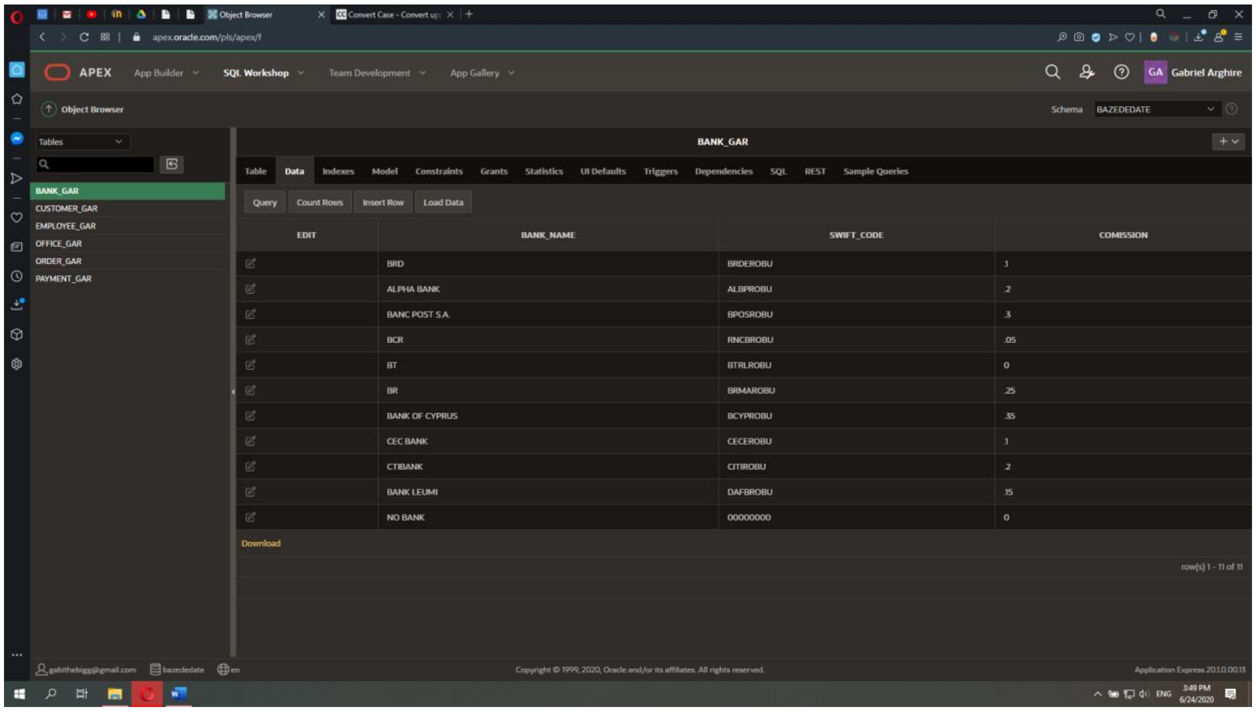
row(s) 16 - 18 of 18

18 Statements Processed

18 Successful

0 With Errors

Data from ”BANK_GAR” table

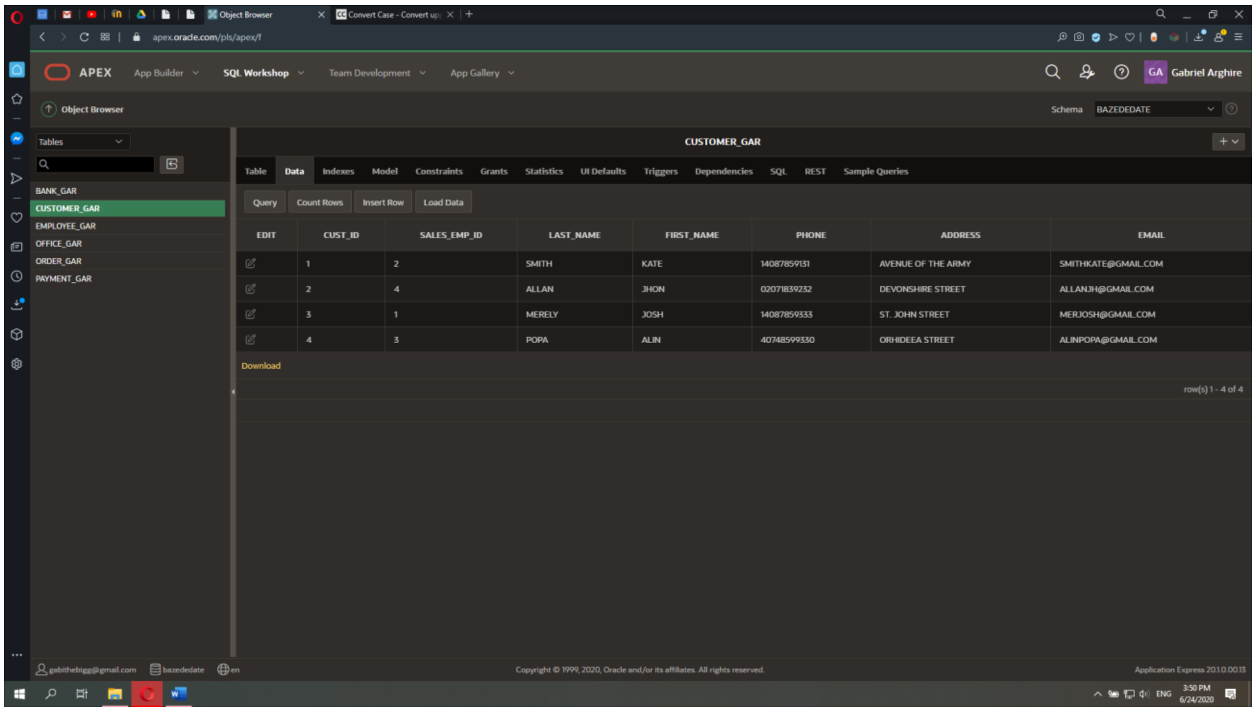


BANK_GAR

EDIT	BANK_NAME	SWIFT_CODE	COMISSION
	BRD	BRDEROBU	.1
	ALPHA BANK	ALBPROBU	.2
	BANC POST S.A.	BPOSROBU	.3
	BCR	RNCBROBU	.05
	BT	BTRLROBU	0
	BR	BRMAROBU	.25
	BANK OF CYPRUS	BCYPROBU	.35
	CEC BANK	CECEROBU	.1
	CTIBANK	CITIROBU	.2
	BANK LEUMI	DAFBROBU	.15
	NO BANK	00000000	0

row(s) 1 - 11 of 11

Data from “CUSTOMER_GAR” table

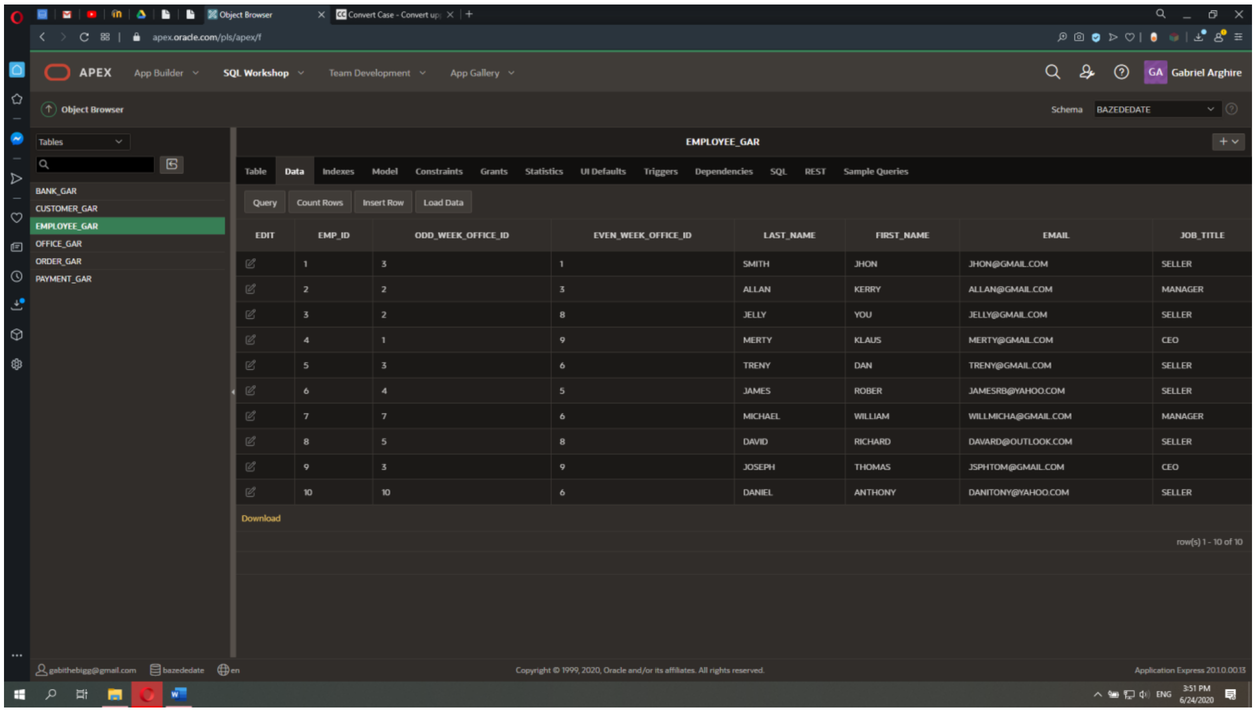


CUSTOMER_GAR

EDIT	CUST_ID	SALES_EMP_ID	LAST_NAME	FIRST_NAME	PHONE	ADDRESS	EMAIL
	1	2	SMITH	KATE	14087859131	AVENUE OF THE ARMY	SMITHKATE@GMAIL.COM
	2	4	ALLAN	JHON	02071839232	DEVONSHIRE STREET	ALLANJH@GMAIL.COM
	3	1	MERELY	JOSH	14087859333	ST. JOHN STREET	MERJOSH@GMAIL.COM
	4	3	POPA	ALIN	40748599330	ORHIDEEA STREET	ALINPOPA@GMAIL.COM

row(s) 1 - 4 of 4

Data from “EMPLOYEE_GAR” table

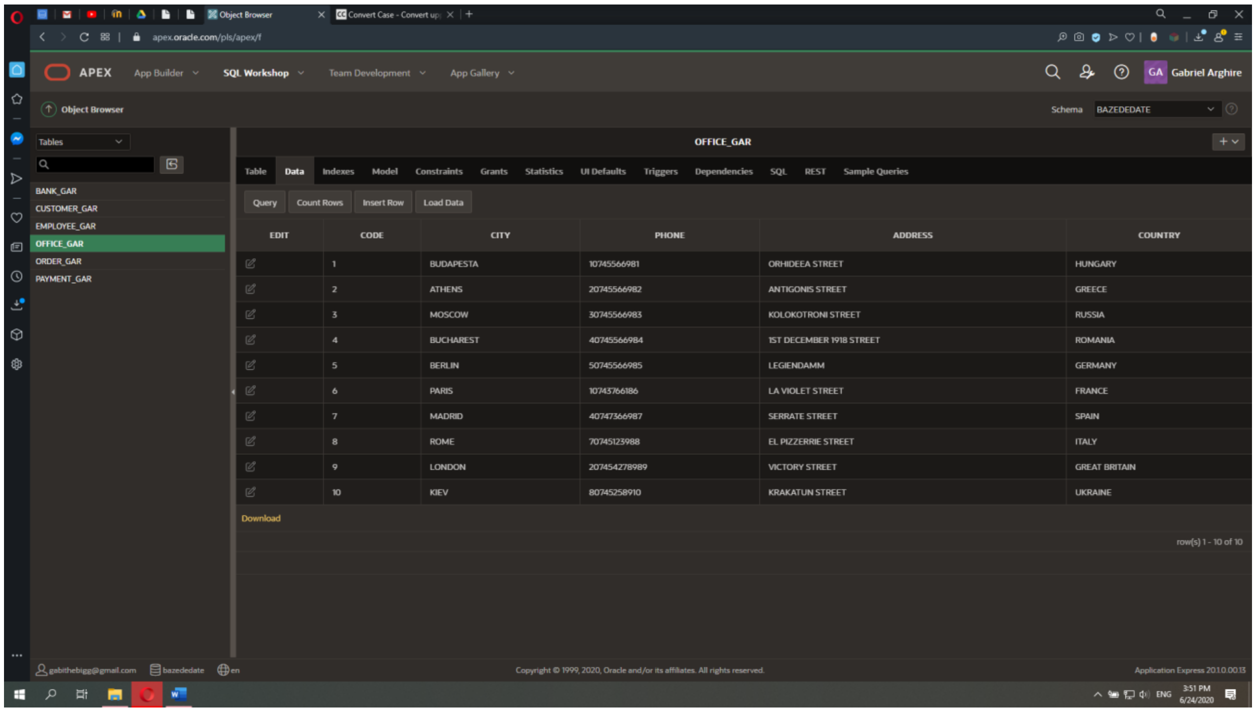


EMPLOYEE_GAR

EDIT	EMP_ID	ODD_WEEK_OFFICE_ID	EVEN_WEEK_OFFICE_ID	LAST_NAME	FIRST_NAME	EMAIL	JOB_TITLE
	1	3	1	SMITH	JHON	JHON@GMAIL.COM	SELLER
	2	2	3	ALLAN	KERRY	ALLAN@GMAIL.COM	MANAGER
	3	2	8	JELLY	YOU	JELLY@GMAIL.COM	SELLER
	4	1	9	MERTY	KLAUS	MERTY@GMAIL.COM	CEO
	5	3	6	TRENY	DAN	TRENY@GMAIL.COM	SELLER
	6	4	5	JAMES	ROBER	JAMESRB@YAHOO.COM	SELLER
	7	7	6	MICHAEL	WILLIAM	WILLMICHA@GMAIL.COM	MANAGER
	8	5	8	DAVID	RICHARD	DAVARD@OUTLOOK.COM	SELLER
	9	3	9	JOSEPH	THOMAS	JSPHTOM@GMAIL.COM	CEO
	10	10	6	DANIEL	ANTHONY	DANITONY@YAHOO.COM	SELLER

row(s) 1 - 10 of 10

Data from “OFFICE_GAR” table

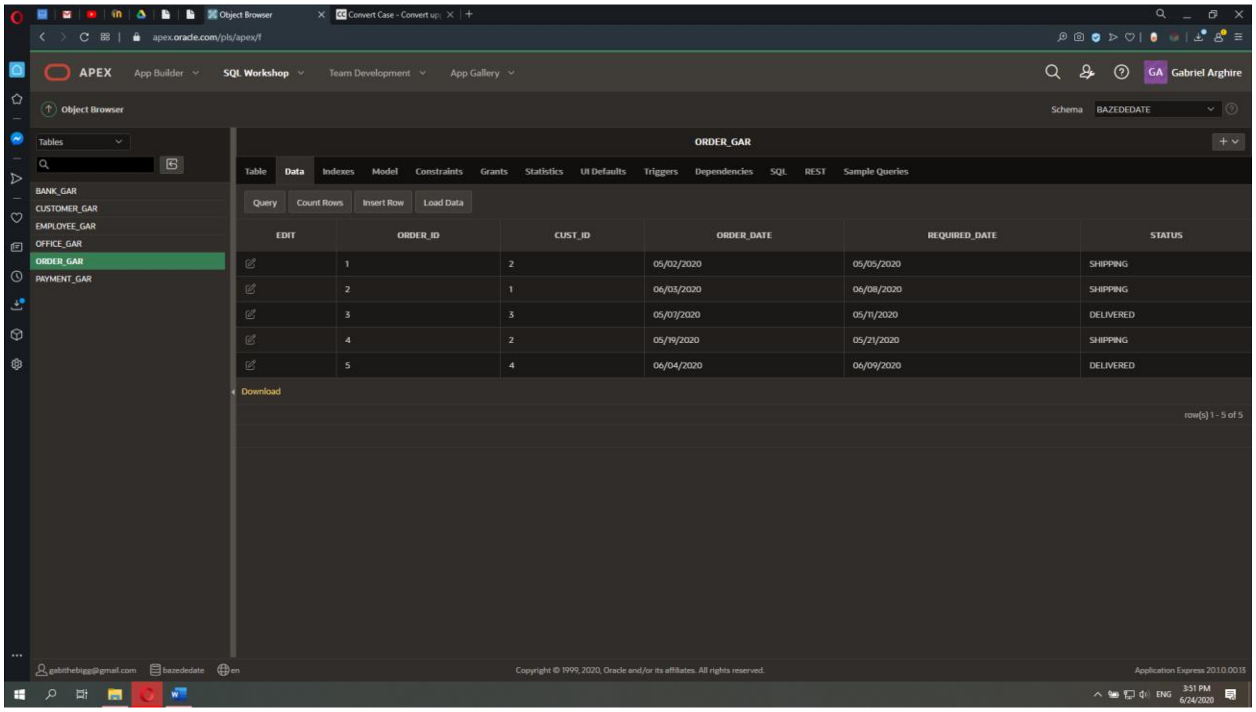


OFFICE_GAR

EDIT	CODE	CITY	PHONE	ADDRESS	COUNTRY
	1	BUDAPESTA	10745566981	ORHIDEEA STREET	HUNGARY
	2	ATHENS	20745566982	ANTIGONIS STREET	GREECE
	3	MOSCOW	30745566983	KOLOKOTRONI STREET	RUSSIA
	4	BUCHAREST	40745566984	1ST DECEMBER 1918 STREET	ROMANIA
	5	BERLIN	50745566985	LEGIENDAMM	GERMANY
	6	PARIS	10743766186	LA VIOLET STREET	FRANCE
	7	MADRID	40747366987	SERRATE STREET	SPAIN
	8	ROME	70745123988	EL PIZZERRIE STREET	ITALY
	9	LONDON	207454278989	VICTORY STREET	GREAT BRITAIN
	10	KIEV	80745258910	KRAKATUN STREET	UKRAINE

row(s) 1 - 10 of 10

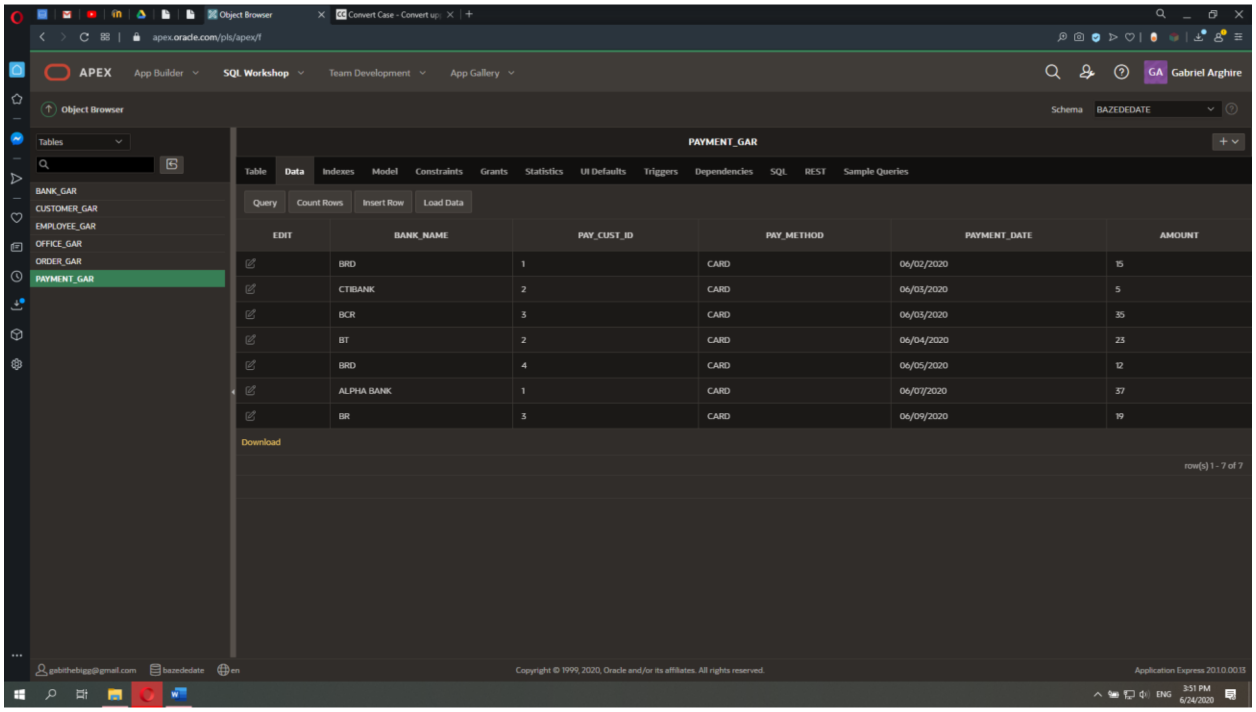
Data from “ORDER_GAR” table


ORDER_GAR

EDIT	ORDER_ID	CUST_ID	ORDER_DATE	REQUIRED_DATE	STATUS
	1	2	05/02/2020	05/05/2020	SHIPPING
	2	1	06/03/2020	06/08/2020	SHIPPING
	3	3	05/07/2020	05/11/2020	DELIVERED
	4	2	05/19/2020	05/21/2020	SHIPPING
	5	4	06/04/2020	06/09/2020	DELIVERED

row(s) 1 - 5 of 5

Data from “PAYMENT_GAR” table


PAYMENT_GAR

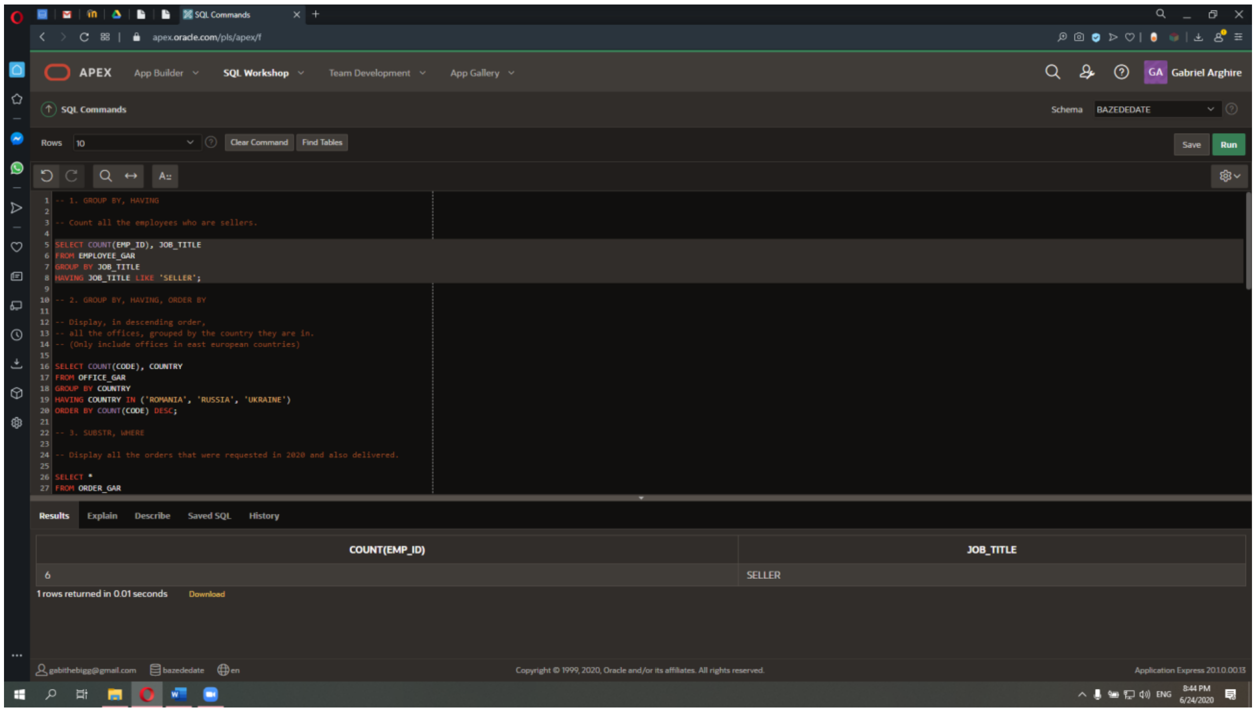
EDIT	BANK_NAME	PAY_CUST_ID	PAY_METHOD	PAYMENT_DATE	AMOUNT
	BRD	1	CARD	06/02/2020	15
	CTIBANK	2	CARD	06/03/2020	5
	BCR	3	CARD	06/03/2020	35
	BT	2	CARD	06/04/2020	23
	BRD	4	CARD	06/05/2020	12
	ALPHA BANK	1	CARD	06/07/2020	37
	BR	3	CARD	06/09/2020	19

row(s) 1 - 7 of 7

Until here I have shown results of ”SCHEMA” script.
Next pages are about displaying the results of the queries from the ”SQL” script.

-- 1. GROUP BY, HAVING, COUNT

-- Count all the employees who are sellers.



```
-- 1. GROUP BY, HAVING

-- Count all the employees who are sellers.

SELECT COUNT(EMP_ID), JOB_TITLE
FROM EMPLOYEE_GAR
GROUP BY JOB_TITLE
HAVING JOB_TITLE LIKE 'SELLER';

-- 2. GROUP BY, HAVING, ORDER BY

-- Display, in descending order,
-- all the offices, grouped by the country they are in.
-- (Only include offices in east european countries)

SELECT COUNT(CODE), COUNTRY
FROM OFFICE_GAR
GROUP BY COUNTRY
HAVING COUNTRY IN ('ROMANIA', 'RUSSIA', 'UKRAINE')
ORDER BY COUNT(CODE) DESC;

-- 3. SUBSTR, WHERE

-- Display all the orders that were requested in 2020 and also delivered.

SELECT *
FROM ORDER_GAR
```

COUNT(EMP_ID)	JOB_TITLE
6	SELLER

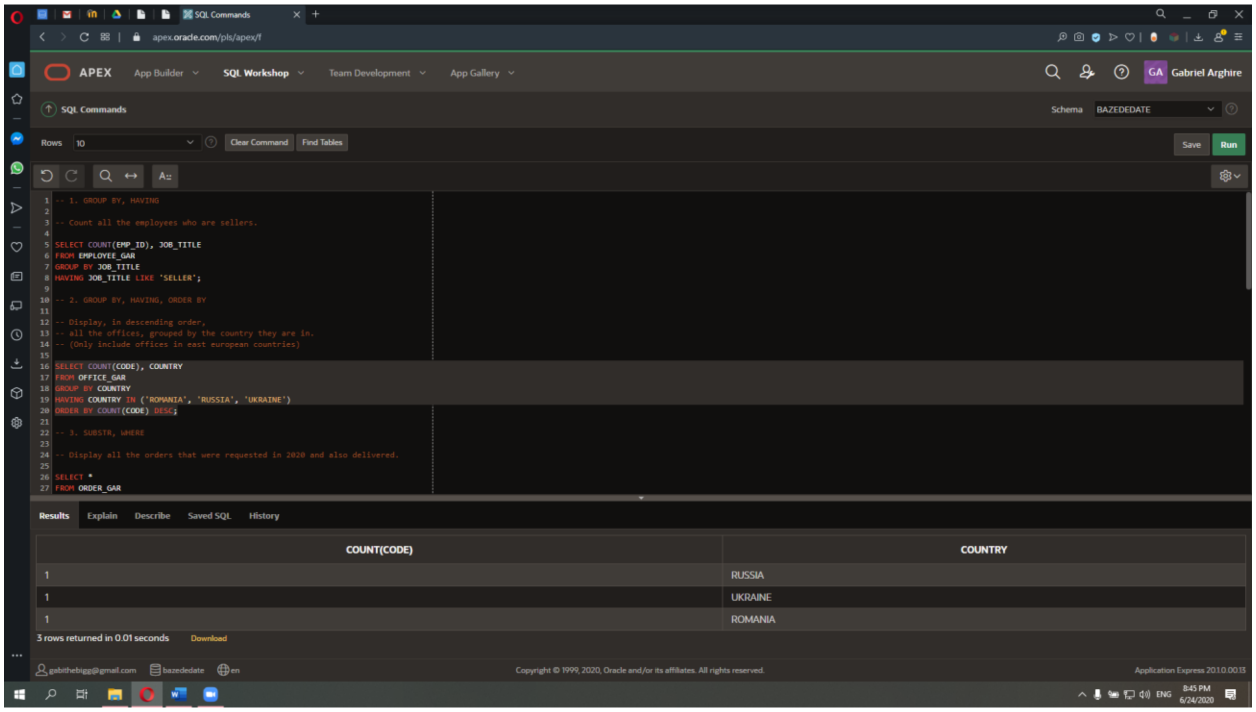
1 rows returned in 0.01 seconds

-- 2. GROUP BY, HAVING, ORDER BY

-- Display, in descending order,

-- all the offices, grouped by the country they are in.

-- (Only include offices in east european countries)



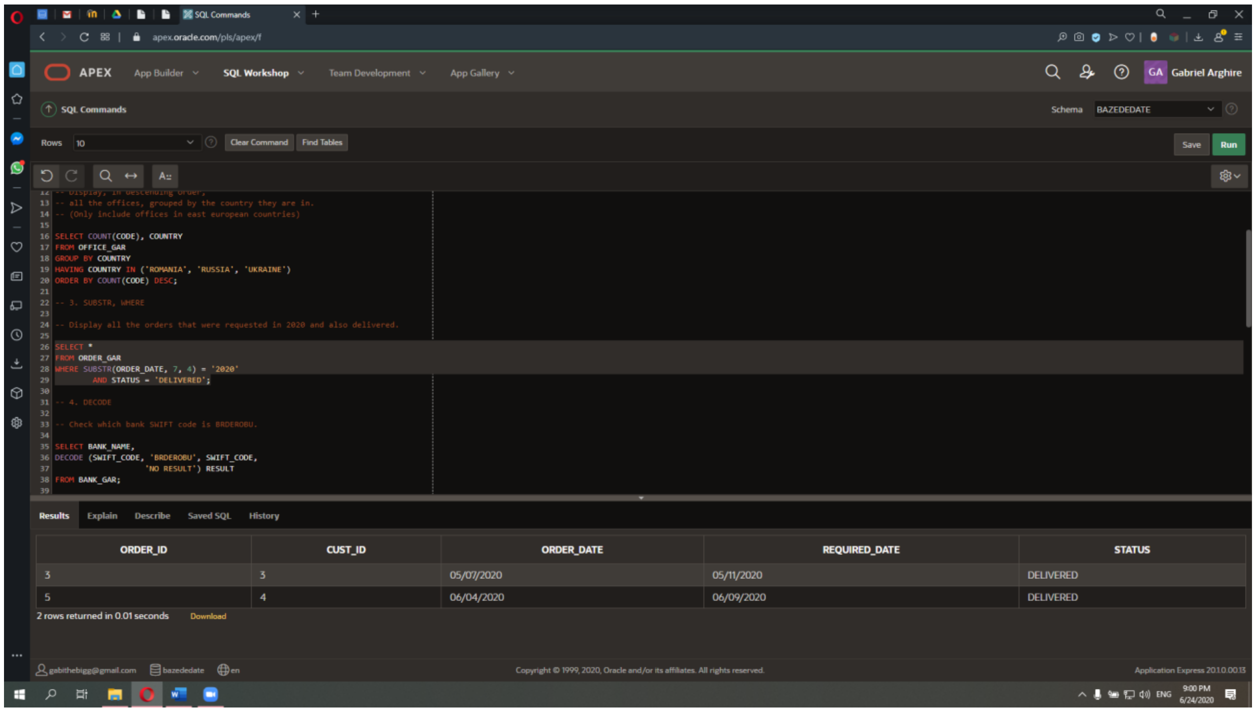
```
-- 1. GROUP BY, HAVING

-- Count all the employees who are sellers.

SELECT COUNT(EMP_ID), JOB_TITLE
FROM EMPLOYEE_GAR
GROUP BY JOB_TITLE
HAVING JOB_TITLE LIKE 'SELLER';

-- 2. GROUP BY, HAVING, ORDER BY

-- Display, in descending order,
-- all the offices, grouped by the country they are in.
-- (Only include offices in east european countries)

SELECT COUNT(CODE), COUNTRY
FROM OFFICE_GAR
GROUP BY COUNTRY
HAVING COUNTRY IN ('ROMANIA', 'RUSSIA', 'UKRAINE')
ORDER BY COUNT(CODE) DESC;

-- 3. SUBSTR, WHERE

-- Display all the orders that were requested in 2020 and also delivered.

SELECT *
FROM ORDER_GAR
```

COUNT(CODE)	COUNTRY
1	RUSSIA
1	UKRAINE
1	ROMANIA

3 rows returned in 0.01 seconds

-- 3. SUBSTR, WHERE

-- Display all the orders that were requested in 2020 and also delivered.



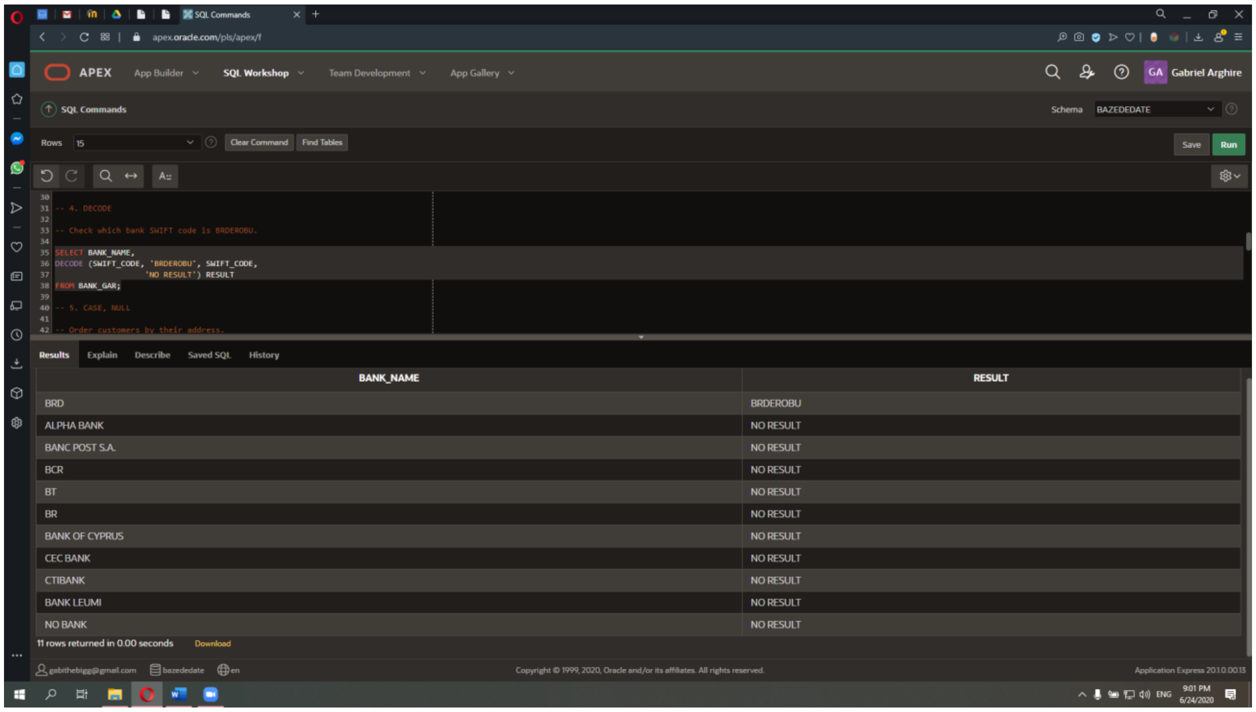
```sql
-- Display, in descending order,
-- all the offices, grouped by the country they are in.
-- (Only include offices in east european countries)

SELECT COUNT(CODE), COUNTRY
FROM OFFICE_GAR
GROUP BY COUNTRY
HAVING COUNTRY IN ('ROMANIA', 'RUSSIA', 'UKRAINE')
ORDER BY COUNT(CODE) DESC;

-- 3. SUBSTR, WHERE

-- Display all the orders that were requested in 2020 and also delivered.

SELECT *
FROM ORDER_GAR
WHERE SUBSTR(ORDER_DATE, 7, 4) = '2020'
        AND STATUS = 'DELIVERED';

-- 4. DECODE

-- Check which bank SWIFT code is BRDEROBU.

SELECT BANK_NAME,
DECODE (SWIFT_CODE, 'BRDEROBU', SWIFT_CODE,
                    'NO RESULT') RESULT
FROM BANK_GAR;
```

ORDER_ID	CUST_ID	ORDER_DATE	REQUIRED_DATE	STATUS
3	3	05/07/2020	05/11/2020	DELIVERED
5	4	06/04/2020	06/09/2020	DELIVERED

2 rows returned in 0.01 seconds Download

-- 4. DECODE

-- Check which bank SWIFT code is BRDEROBU.



```sql
-- 4. DECODE

-- Check which bank SWIFT code is BRDEROBU.

SELECT BANK_NAME,
DECODE (SWIFT_CODE, 'BRDEROBU', SWIFT_CODE,
                    'NO RESULT') RESULT
FROM BANK_GAR;

-- 5. CASE, NULL

-- Order customers by their address.
```

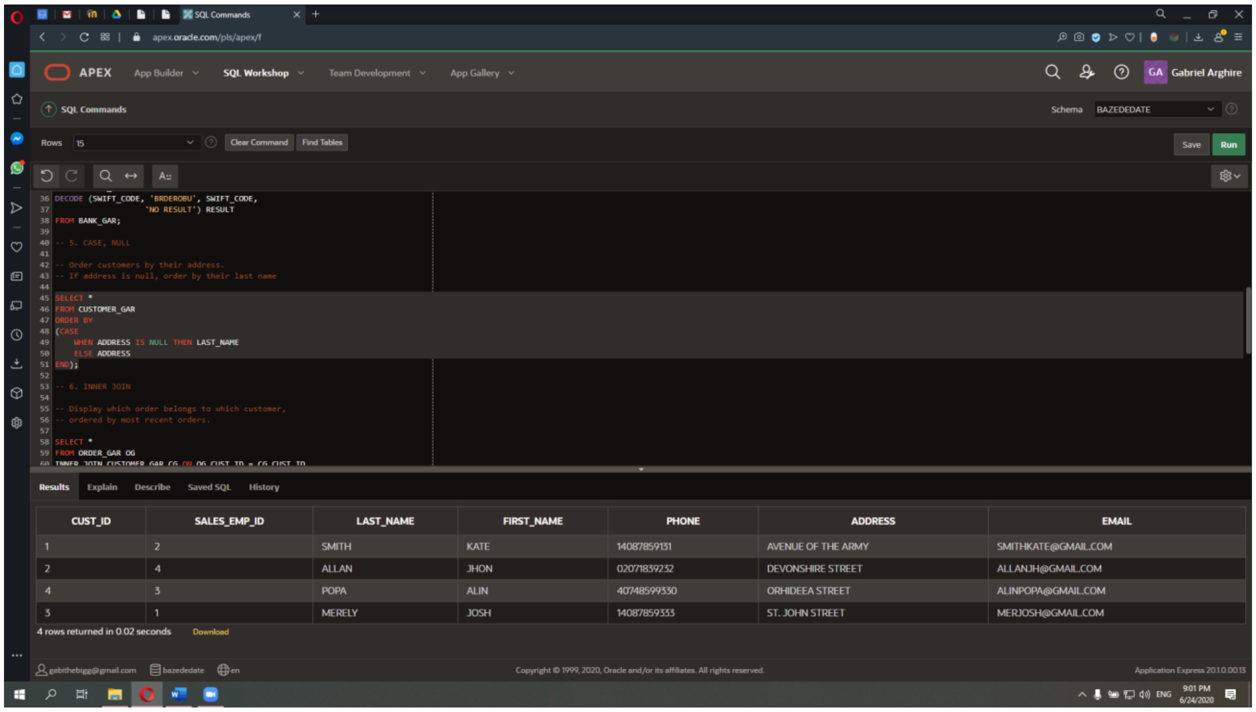
BANK_NAME	RESULT
BRD	BRDEROBU
ALPHA BANK	NO RESULT
BANC POST S.A.	NO RESULT
BCR	NO RESULT
BT	NO RESULT
BR	NO RESULT
BANK OF CYPRUS	NO RESULT
CEC BANK	NO RESULT
CITIBANK	NO RESULT
BANK LEUMI	NO RESULT
NO BANK	NO RESULT

11 rows returned in 0.00 seconds Download

-- 5. CASE, NULL

-- Order customers by their address.

-- If address is null, order by their last name



```
DECODE (SWIFT_CODE, 'BRDEROBU', SWIFT_CODE,
                    'NO RESULT') RESULT
FROM BANK_GAR;

-- 5. CASE, NULL

-- Order customers by their address.
-- If address is null, order by their last name

SELECT *
FROM CUSTOMER_GAR
ORDER BY
(CASE
    WHEN ADDRESS IS NULL THEN LAST_NAME
    ELSE ADDRESS
END);

-- 6. INNER JOIN

-- Display which order belongs to which customer,
-- ordered by most recent orders.

SELECT *
FROM ORDER_GAR OG
INNER JOIN CUSTOMER_GAR CG ON OG.CUST_ID = CG.CUST_ID
```

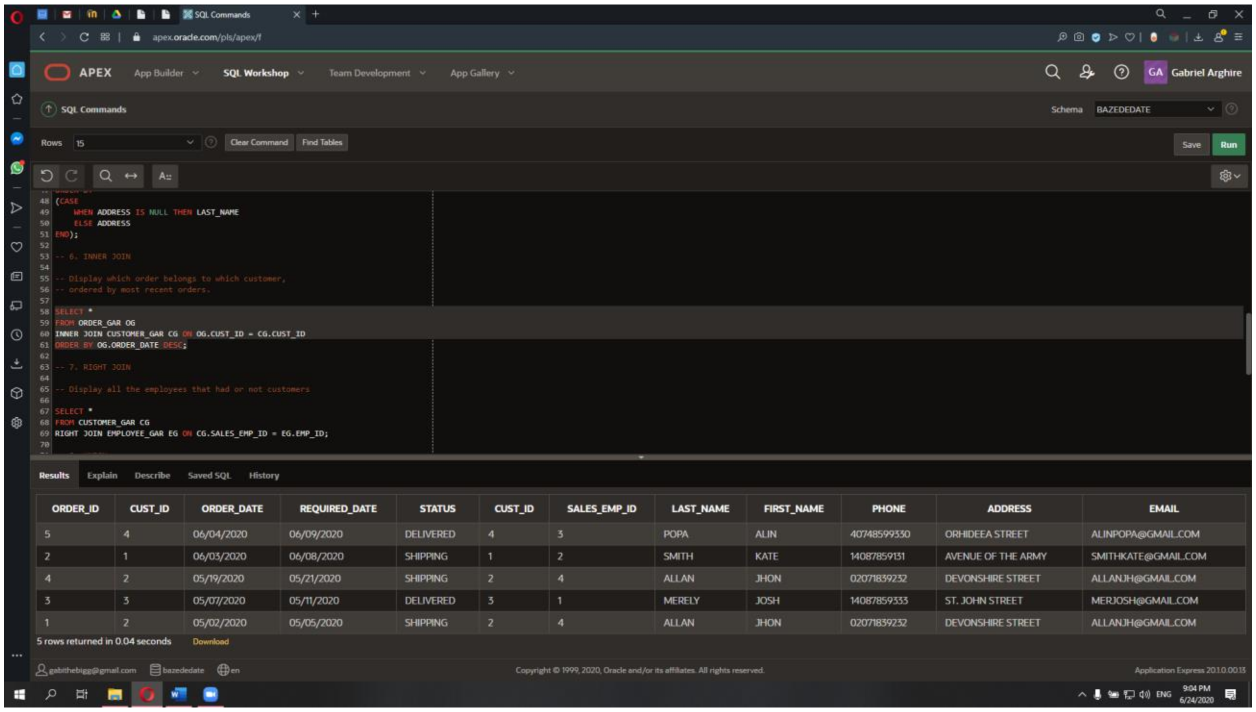
CUST_ID	SALES_EMP_ID	LAST_NAME	FIRST_NAME	PHONE	ADDRESS	EMAIL
1	2	SMITH	KATE	14087859131	AVENUE OF THE ARMY	SMITHKATE@GMAIL.COM
2	4	ALLAN	JHON	02071839232	DEVONSHIRE STREET	ALLANJH@GMAIL.COM
4	3	POPA	ALIN	40748599330	ORHIDEEA STREET	ALINPOPA@GMAIL.COM
3	1	MERELY	JOSH	14087859333	ST. JOHN STREET	MERJOSH@GMAIL.COM

4 rows returned in 0.02 seconds

-- 6. INNER JOIN

-- Display which order belongs to which customer,

-- ordered by most recent orders.



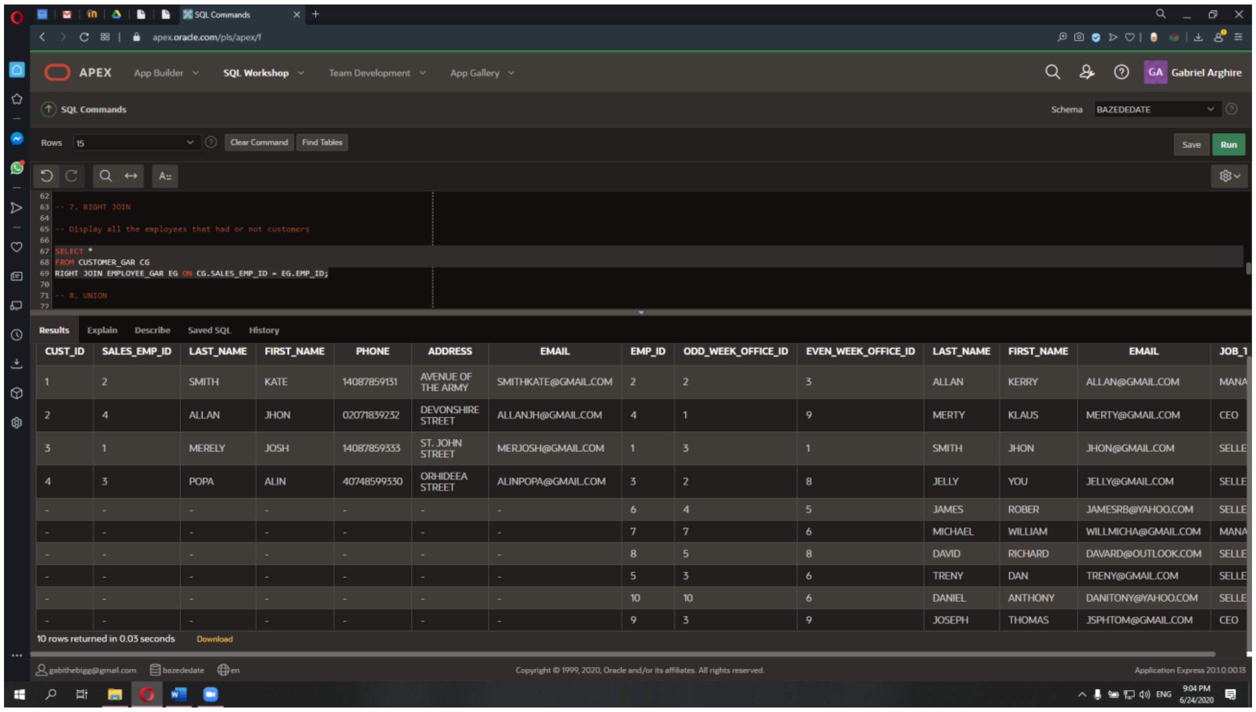
```
(CASE
    WHEN ADDRESS IS NULL THEN LAST_NAME
    ELSE ADDRESS
END);

-- 6. INNER JOIN

-- Display which order belongs to which customer,
-- ordered by most recent orders.

SELECT *
FROM ORDER_GAR OG
INNER JOIN CUSTOMER_GAR CG ON OG.CUST_ID = CG.CUST_ID
ORDER BY OG.ORDER_DATE DESC;

-- 7. RIGHT JOIN

-- Display all the employees that had or not customers

SELECT *
FROM CUSTOMER_GAR CG
RIGHT JOIN EMPLOYEE_GAR EG ON CG.SALES_EMP_ID = EG.EMP_ID;
```

ORDER_ID	CUST_ID	ORDER_DATE	REQUIRED_DATE	STATUS	CUST_ID	SALES_EMP_ID	LAST_NAME	FIRST_NAME	PHONE	ADDRESS	EMAIL
5	4	06/04/2020	06/09/2020	DELIVERED	4	3	POPA	ALIN	40748599330	ORHIDEEA STREET	ALINPOPA@GMAIL.COM
2	1	06/03/2020	06/08/2020	SHIPPING	1	2	SMITH	KATE	14087859131	AVENUE OF THE ARMY	SMITHKATE@GMAIL.COM
4	2	05/19/2020	05/21/2020	SHIPPING	2	4	ALLAN	JHON	02071839232	DEVONSHIRE STREET	ALLANJH@GMAIL.COM
3	3	05/07/2020	05/11/2020	DELIVERED	3	1	MERELY	JOSH	14087859333	ST. JOHN STREET	MERJOSH@GMAIL.COM
1	2	05/02/2020	05/05/2020	SHIPPING	2	4	ALLAN	JHON	02071839232	DEVONSHIRE STREET	ALLANJH@GMAIL.COM

5 rows returned in 0.04 seconds

-- 7. RIGHT JOIN

-- Display all the employees that had or not customers



```
-- 7. RIGHT JOIN

-- Display all the employees that had or not customers

SELECT *
FROM CUSTOMER_GAR CG
RIGHT JOIN EMPLOYEE_GAR EG ON CG.SALES_EMP_ID = EG.EMP_ID;

-- 8. UNION
```

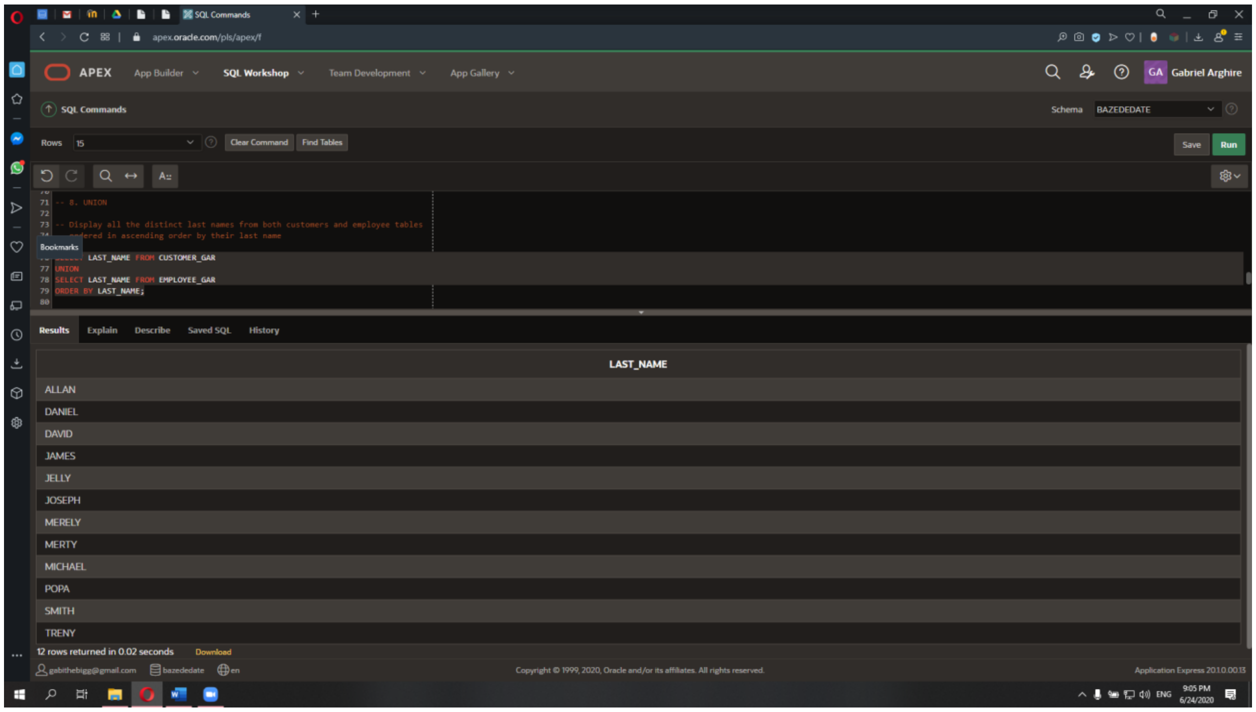
CUST_ID	SALES_EMP_ID	LAST_NAME	FIRST_NAME	PHONE	ADDRESS	EMAIL	EMP_ID	ODD_WEEK_OFFICE_ID	EVEN_WEEK_OFFICE_ID	LAST_NAME	FIRST_NAME	EMAIL	JOB_T
1	2	SMITH	KATE	14087859131	AVENUE OF THE ARMY	SMITHKATE@GMAIL.COM	2	2	3	ALLAN	KERRY	ALLAN@GMAIL.COM	MANA
2	4	ALLAN	JHON	02071839232	DEVONSHIRE STREET	ALLANJH@GMAIL.COM	4	1	9	MERTY	KLAUS	MERTY@GMAIL.COM	CEO
3	1	MERELY	JOSH	14087859333	ST. JOHN STREET	MERJOSH@GMAIL.COM	1	3	1	SMITH	JHON	JHON@GMAIL.COM	SELLE
4	3	POPA	ALIN	40748599330	ORHIDEEA STREET	ALINPOPA@GMAIL.COM	3	2	8	JELLY	YOU	JELLY@GMAIL.COM	SELLE
-	-	-	-	-	-	-	6	4	5	JAMES	ROBER	JAMESRB@YAHOO.COM	SELLE
-	-	-	-	-	-	-	7	7	6	MICHAEL	WILLIAM	WILLMICHA@GMAIL.COM	MANA
-	-	-	-	-	-	-	8	5	8	DAVID	RICHARD	DAVARD@OUTLOOK.COM	SELLE
-	-	-	-	-	-	-	5	3	6	TRENY	DAN	TRENY@GMAIL.COM	SELLE
-	-	-	-	-	-	-	10	10	6	DANIEL	ANTHONY	DANITONY@YAHOO.COM	SELLE
-	-	-	-	-	-	-	9	3	9	JOSEPH	THOMAS	JSPHTOM@GMAIL.COM	CEO

10 rows returned in 0.03 seconds

-- 8. UNION

-- Display all the distinct last names from both customers and employee tables

-- ordered in ascending order by their last name



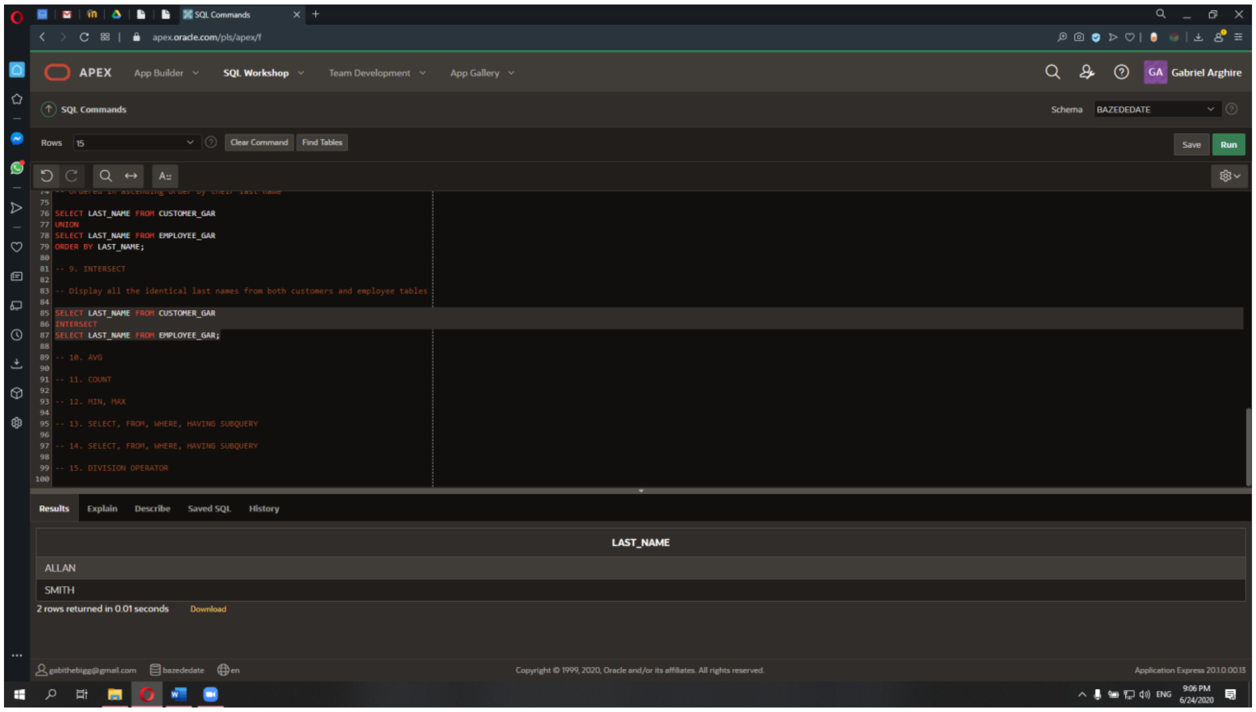
```
-- 8. UNION

-- Display all the distinct last names from both customers and employee tables
-- ordered in ascending order by their last name

SELECT LAST_NAME FROM CUSTOMER_GAR
UNION
SELECT LAST_NAME FROM EMPLOYEE_GAR
ORDER BY LAST_NAME;
```

LAST_NAME
ALLAN
DANIEL
DAVID
JAMES
JELLY
JOSEPH
MERELY
MERTY
MICHAEL
POPA
SMITH
TRENY

12 rows returned in 0.02 seconds

-- 9. INTERSECT

-- Display all the identical last names from both customers and employee tables



```
-- ordered in ascending order by their last name

SELECT LAST_NAME FROM CUSTOMER_GAR
UNION
SELECT LAST_NAME FROM EMPLOYEE_GAR
ORDER BY LAST_NAME;

-- 9. INTERSECT

-- Display all the identical last names from both customers and employee tables

SELECT LAST_NAME FROM CUSTOMER_GAR
INTERSECT
SELECT LAST_NAME FROM EMPLOYEE_GAR;

-- 10. AVG

-- 11. COUNT

-- 12. MIN, MAX

-- 13. SELECT, FROM, WHERE, HAVING SUBQUERY

-- 14. SELECT, FROM, WHERE, HAVING SUBQUERY

-- 15. DIVISION OPERATOR
```

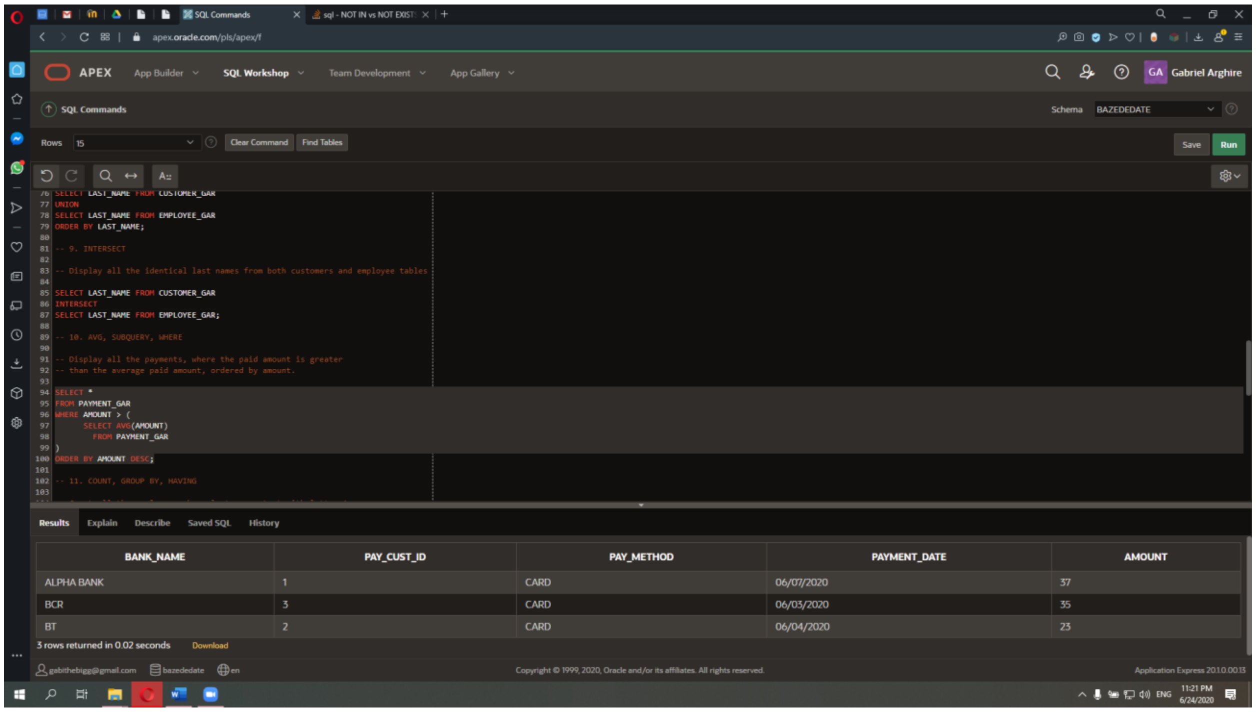
LAST_NAME
ALLAN
SMITH

2 rows returned in 0.01 seconds

-- 10. AVG, SUBQUERY, WHERE

-- Display all the payments, where the paid amount is greater

-- than the average paid amount, ordered by amount.



```
SELECT LAST_NAME FROM CUSTOMER_GAR
UNION
SELECT LAST_NAME FROM EMPLOYEE_GAR
ORDER BY LAST_NAME;

-- 9. INTERSECT

-- Display all the identical last names from both customers and employee tables

SELECT LAST_NAME FROM CUSTOMER_GAR
INTERSECT
SELECT LAST_NAME FROM EMPLOYEE_GAR;

-- 10. AVG, SUBQUERY, WHERE

-- Display all the payments, where the paid amount is greater
-- than the average paid amount, ordered by amount.

SELECT *
FROM PAYMENT_GAR
WHERE AMOUNT > (
      SELECT AVG(AMOUNT)
        FROM PAYMENT_GAR
)
ORDER BY AMOUNT DESC;

-- 11. COUNT, GROUP BY, HAVING
```

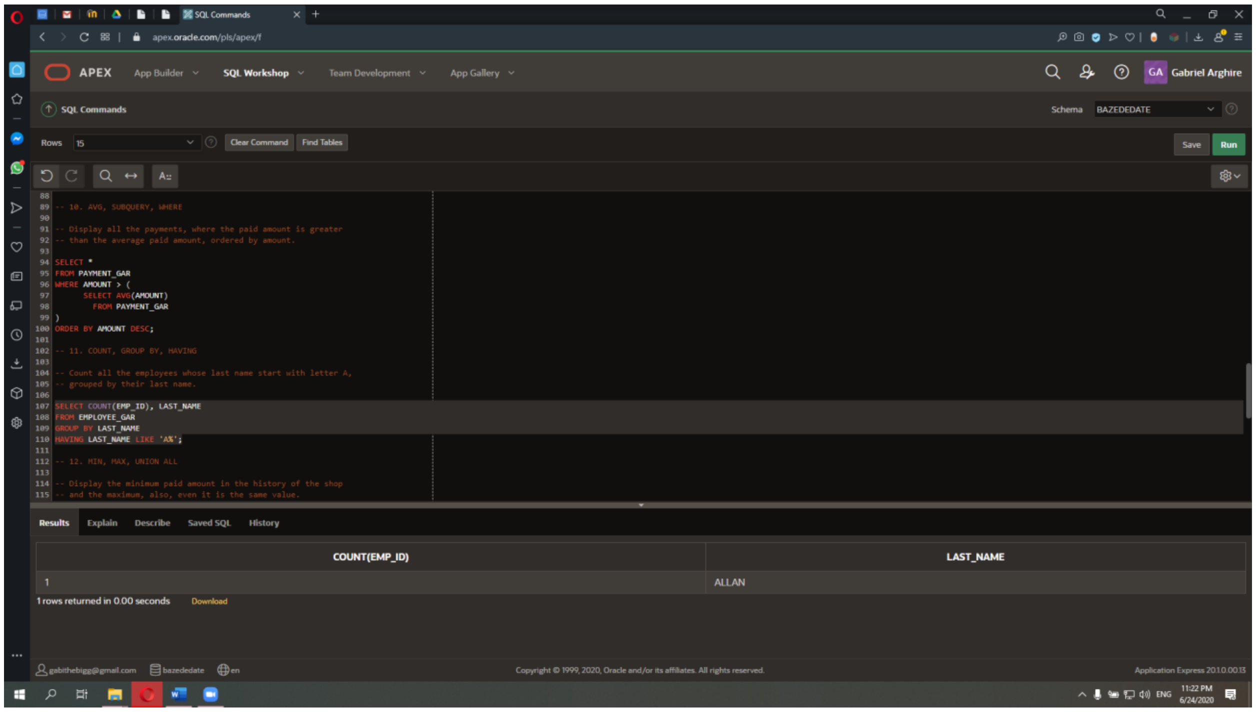
BANK_NAME	PAY_CUST_ID	PAY_METHOD	PAYMENT_DATE	AMOUNT
ALPHA BANK	1	CARD	06/07/2020	37
BCR	3	CARD	06/03/2020	35
BT	2	CARD	06/04/2020	23

3 rows returned in 0.02 seconds

-- 11. COUNT, GROUP BY, HAVING

-- Count all the employees whose last name start with letter A,

-- grouped by their last name.



```sql

-- 10. AVG, SUBQUERY, WHERE

-- Display all the payments, where the paid amount is greater
-- than the average paid amount, ordered by amount.

SELECT *
FROM PAYMENT_GAR
WHERE AMOUNT > (
      SELECT AVG(AMOUNT)
        FROM PAYMENT_GAR
)
ORDER BY AMOUNT DESC;

-- 11. COUNT, GROUP BY, HAVING

-- Count all the employees whose last name start with letter A,
-- grouped by their last name.

SELECT COUNT(EMP_ID), LAST_NAME
FROM EMPLOYEE_GAR
GROUP BY LAST_NAME
HAVING LAST_NAME LIKE 'A%';

-- 12. MIN, MAX, UNION ALL

-- Display the minimum paid amount in the history of the shop
-- and the maximum, also, even it is the same value.
```

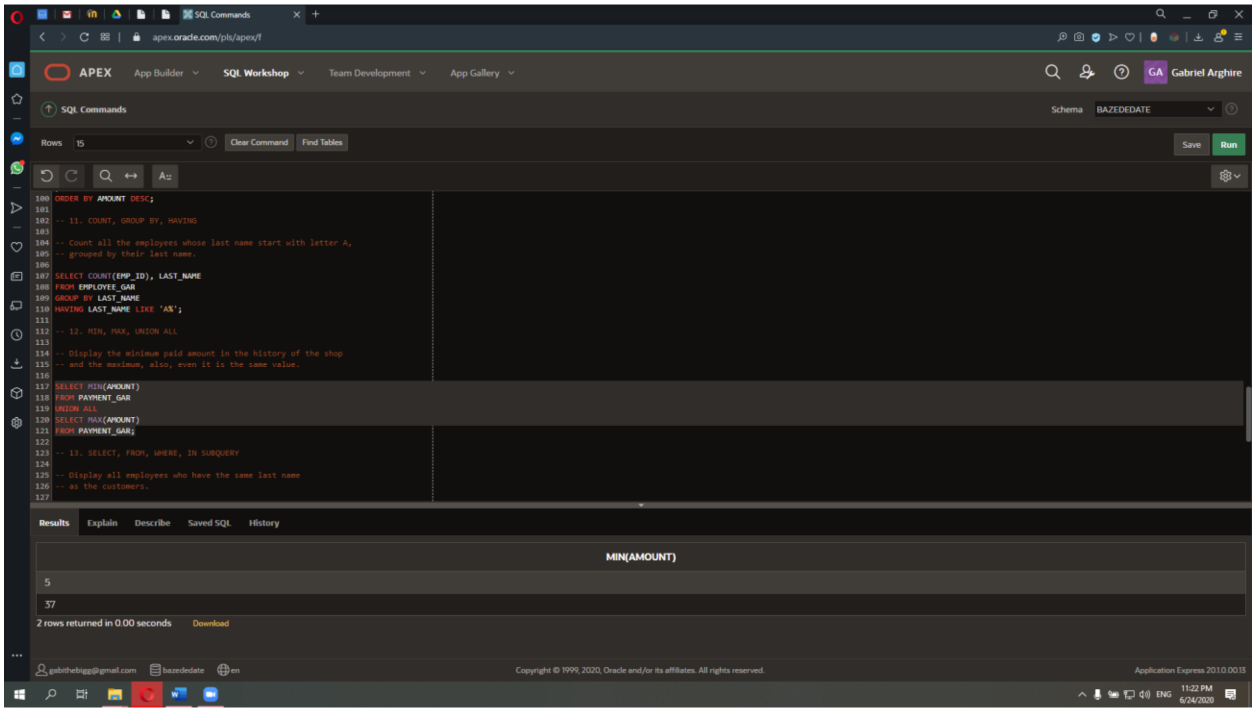
COUNT(EMP_ID)	LAST_NAME
1	ALLAN

1 rows returned in 0.00 seconds

-- 12. MIN, MAX, UNION ALL

-- Display the minimum paid amount in the history of the shop

-- and the maximum, also, even it is the same value.



```sql
ORDER BY AMOUNT DESC;

-- 11. COUNT, GROUP BY, HAVING

-- Count all the employees whose last name start with letter A,
-- grouped by their last name.

SELECT COUNT(EMP_ID), LAST_NAME
FROM EMPLOYEE_GAR
GROUP BY LAST_NAME
HAVING LAST_NAME LIKE 'A%';

-- 12. MIN, MAX, UNION ALL

-- Display the minimum paid amount in the history of the shop
-- and the maximum, also, even it is the same value.

SELECT MIN(AMOUNT)
FROM PAYMENT_GAR
UNION ALL
SELECT MAX(AMOUNT)
FROM PAYMENT_GAR;

-- 13. SELECT, FROM, WHERE, IN SUBQUERY

-- Display all employees who have the same last name
-- as the customers.

```

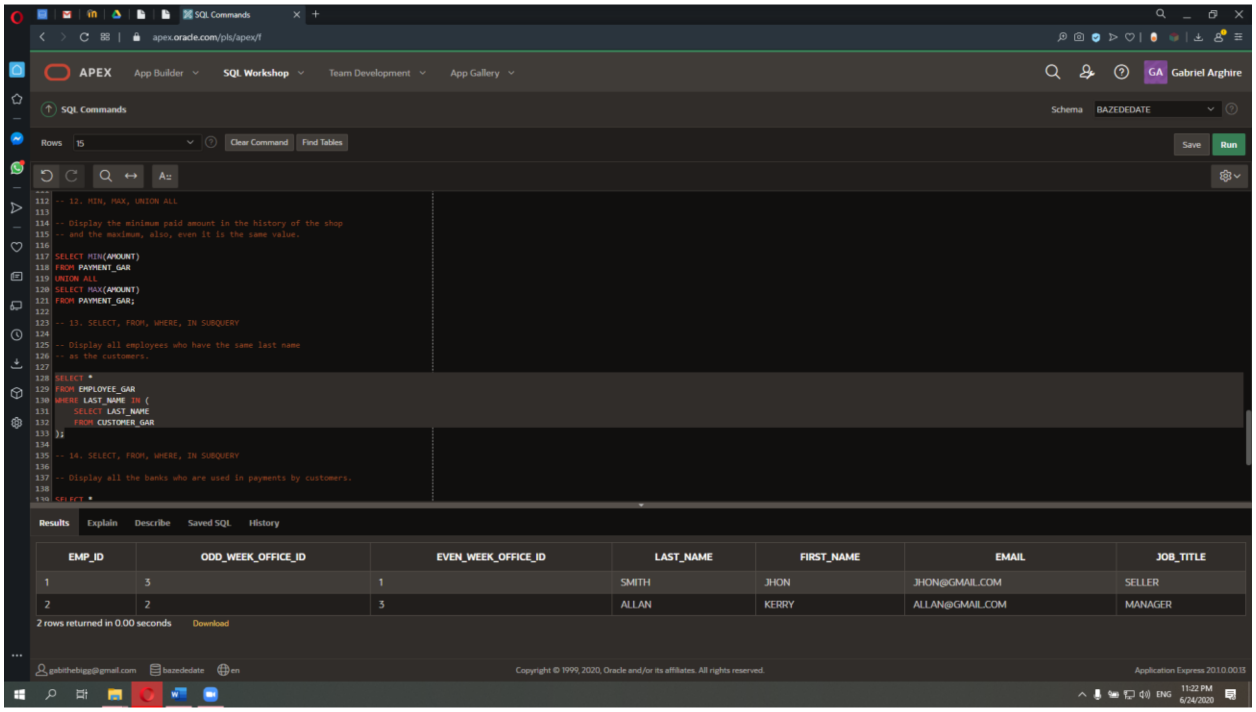
MIN(AMOUNT)
5
37

2 rows returned in 0.00 seconds

-- 13. SELECT, FROM, WHERE, IN SUBQUERY

-- Display all employees who have the same last name

-- as the customers.



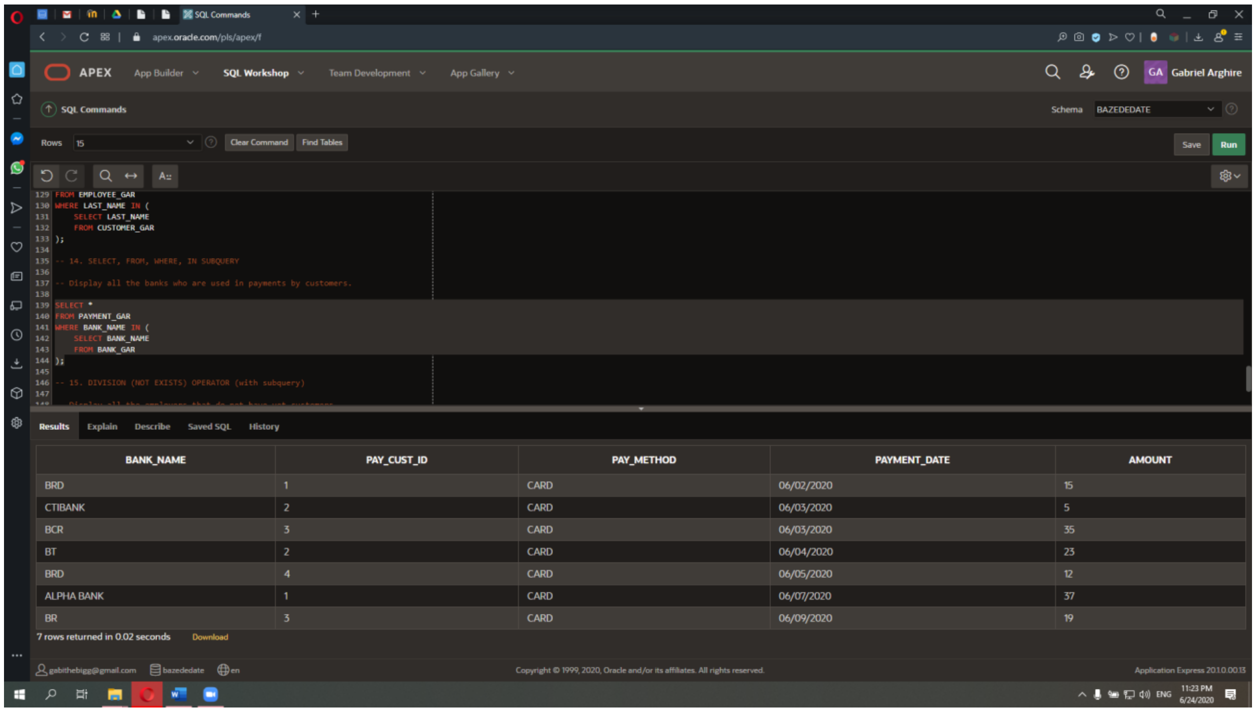
```
-- 12. MIN, MAX, UNION ALL

-- Display the minimum paid amount in the history of the shop
-- and the maximum, also, even it is the same value.

SELECT MIN(AMOUNT)
FROM PAYMENT_GAR
UNION ALL
SELECT MAX(AMOUNT)
FROM PAYMENT_GAR;

-- 13. SELECT, FROM, WHERE, IN SUBQUERY

-- Display all employees who have the same last name
-- as the customers.

SELECT *
FROM EMPLOYEE_GAR
WHERE LAST_NAME IN (
    SELECT LAST_NAME
    FROM CUSTOMER_GAR
);

-- 14. SELECT, FROM, WHERE, IN SUBQUERY

-- Display all the banks who are used in payments by customers.

SELECT *
```

EMP_ID	ODD_WEEK_OFFICE_ID	EVEN_WEEK_OFFICE_ID	LAST_NAME	FIRST_NAME	EMAIL	JOB_TITLE
1	3	1	SMITH	JHON	JHON@GMAIL.COM	SELLER
2	2	3	ALLAN	KERRY	ALLAN@GMAIL.COM	MANAGER

2 rows returned in 0.00 seconds

-- 14. SELECT, FROM, WHERE, IN SUBQUERY

-- Display all the banks who are used in payments by customers.



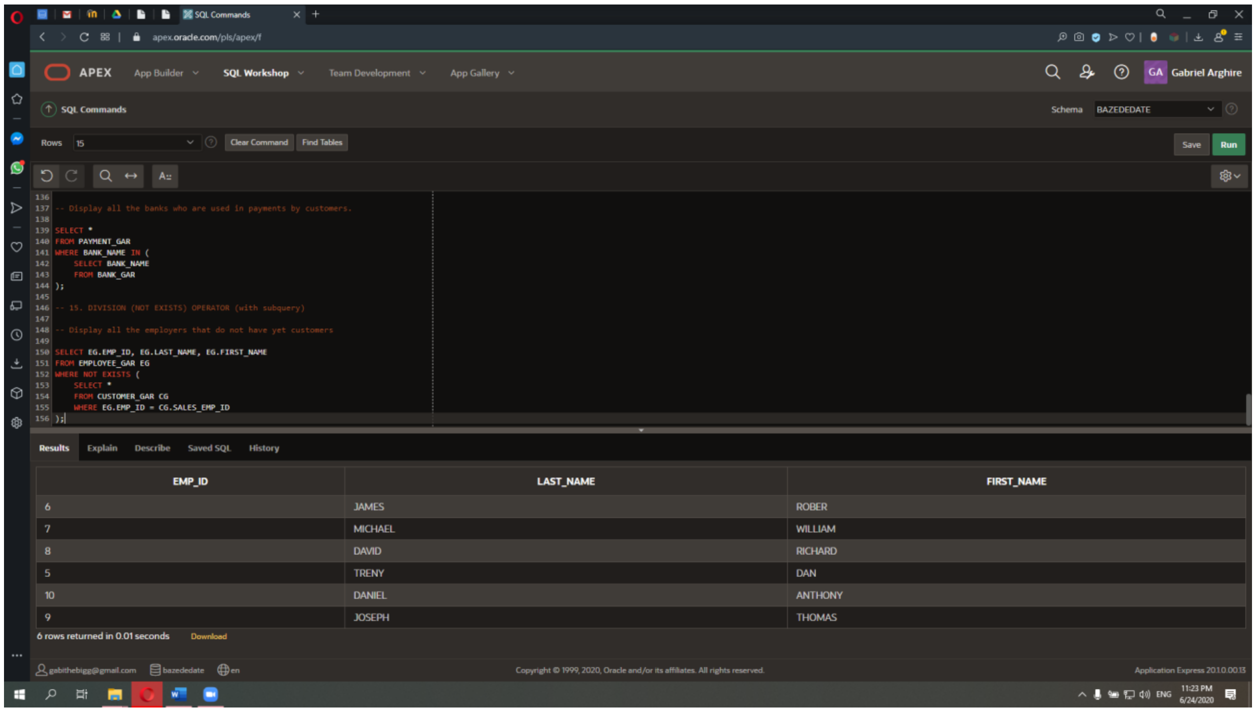
```
FROM EMPLOYEE_GAR
WHERE LAST_NAME IN (
    SELECT LAST_NAME
    FROM CUSTOMER_GAR
);

-- 14. SELECT, FROM, WHERE, IN SUBQUERY

-- Display all the banks who are used in payments by customers.

SELECT *
FROM PAYMENT_GAR
WHERE BANK_NAME IN (
    SELECT BANK_NAME
    FROM BANK_GAR
);

-- 15. DIVISION (NOT EXISTS) OPERATOR (with subquery)
```

BANK_NAME	PAY_CUST_ID	PAY_METHOD	PAYMENT_DATE	AMOUNT
BRD	1	CARD	06/02/2020	15
CTIBANK	2	CARD	06/03/2020	5
BCR	3	CARD	06/03/2020	35
BT	2	CARD	06/04/2020	23
BRD	4	CARD	06/05/2020	12
ALPHA BANK	1	CARD	06/07/2020	37
BR	3	CARD	06/09/2020	19

7 rows returned in 0.02 seconds

-- 15. DIVISION (NOT EXISTS) OPERATOR (with subquery)

-- Display all the employers that do not have yet customers



```
-- Display all the banks who are used in payments by customers.

SELECT *
FROM PAYMENT_GAR
WHERE BANK_NAME IN (
    SELECT BANK_NAME
    FROM BANK_GAR
);

-- 15. DIVISION (NOT EXISTS) OPERATOR (with subquery)

-- Display all the employers that do not have yet customers

SELECT EG.EMP_ID, EG.LAST_NAME, EG.FIRST_NAME
FROM EMPLOYEE_GAR EG
WHERE NOT EXISTS (
    SELECT *
    FROM CUSTOMER_GAR CG
    WHERE EG.EMP_ID = CG.SALES_EMP_ID
);
```

EMP_ID	LAST_NAME	FIRST_NAME
6	JAMES	ROBER
7	MICHAEL	WILLIAM
8	DAVID	RICHARD
5	TRENY	DAN
10	DANIEL	ANTHONY
9	JOSEPH	THOMAS

6 rows returned in 0.01 seconds