Final Report

CPS 510 Database Systems
Prof. Soheila Bashardoust Tajali
December 2, 2021

Members	
Mahad Farah	500783718
Edward Boguslavsky	500884150
Mueeza Ali	500825293

Contents

Assignment 1:

Payroll DBMS

Project Description

(2 - 3 paragraphs)

The payroll system is a necessity for every business to function. The larger a business becomes, the more complex it becomes to pay the employees their fair share. Pages of information must be stored on each employee to guarantee they are securely and consistently receiving the correct amount. Furthermore, this information must be easily accessible to the employee to increase transparency.

Within a payroll system there are two fundamental components, the payer (HR) and the payee. The payee, also known as the employee, using a payroll system will have a series of primary attributes which include: employee ID, Name, Schedule, and Hourly Rate. Supplementary info may include information such as department/position to correspond to an hourly rate. Employees' detailed views hold data like banking info, address, and contact information.

To ensure a smooth procedure and accurate timestamps, employees' schedules will include info like date and the respective clock in and clock out timings for the respective day. In large scale companies, punctuality can be difficult to evaluate hence the benefit of electronic clocking systems. Below you will find the projected DB structure that may be used to implement a PR DBMS.

Possible Queries & Answers

Functions of the System

1. **Which employees have an hourly rate of $24/hr?**
   Aegs Benedict and Hugh Mungus are the only employees that make $24/hr.

2. **How many total hours has Aegs Benedict worked on September 20th?**
   He worked 9 hours on Sept 20th.

3. **List all employees in the Retail Department?**
   Aegs Benedict, Hugh Mungus, and Bob Sicle work in the Retail Department.

4. **Aegs has recently received a promotion, updating his position and hourly rate accordingly.**
   Update Aegs position entry in the database for his promotion to be reflected in all related entries in the database.

5. **How much do I need to pay Aegs for his work on Sept 20th?**

9 ╳ $24 = $216.

Think of different users for the database of Figure 1.2. What type of applications would each user need? To which user category would each belong and what type of interface would they need?

Employee - Clock in / clock out / male time off request

- Clock In / clock out portal
- Personal info/ banking (editable)
- Hourly rate / departmental position / view.

HR - Enters new employees/ set departmental positions/ wages

- Administer any incorrect punches (clock in/ clock out)
- Set department positions and wages

Finance

- Pay out accrual of hours
- Bank connection

Employee

ID	First Name	Last Name	Address	Phone Number	Position
#00004519	Aegs	Benedict	7 Franklin Blvd	(416) 905-1800	Service Technician
#00006815	Hugh	Mungus	20 Solaire Rd	(647) 800-8511	Service Technician
#00007994	Bob	Sicle	1127 Main St	(905) 013-1337	Sales Associate

Schedule

Employee ID: #00004519

Date	Clock In	Clock Out	Start Break	End Break

Sept 20, 2021	8:00	5:00	11:03	12:01
Sept 21, 2021	8:00	6:00	11:00	12:00
Sept 22, 2021	8:00	6:30	11:00	12:04
Sept 23, 2021	8:02	6:02	11:00	12:00
Sept 24, 2021	8:00	5:15	11:01	12:00

Tech Retail Department

Department Positions	Hourly Rate
Service Technicians	$24.00/hr
Assistant Manager	$26.00/hr
Sales Associate	$16.00/hr
Head Manager	$34.00/hr

Financial Account

ID	Account Number	Billing Address	Pay Date (Dynamic)
#00004519	XXXX XXXX XXXX 9823	*[Same as Address]*	September 7th 2021
#00006815	XXXX XXXX XXXX 2490	86 Twenty First St	September 7th 2021
#00007994	XXXX XXXX XXXX 8904	*[Same as Address]*	September 7th 2021

Future “Profound” Possibilities / Rough Work

- Payer Dashboard
- Vacation and Time Off Request
- Province specific tax rates
- Hour Accrual and Pay Date Detailing
- Add more departments

Empoyee

Emp Id	Name	L Name
123	Bob	Joe
456	Frank	Derek

Schedule

Emp Id	Date	Clock In	Clock Out
id	. .		
:			
:			
:			

Employee S tus

Employee	Depart/ positio	ourly	Adress	Active

> department

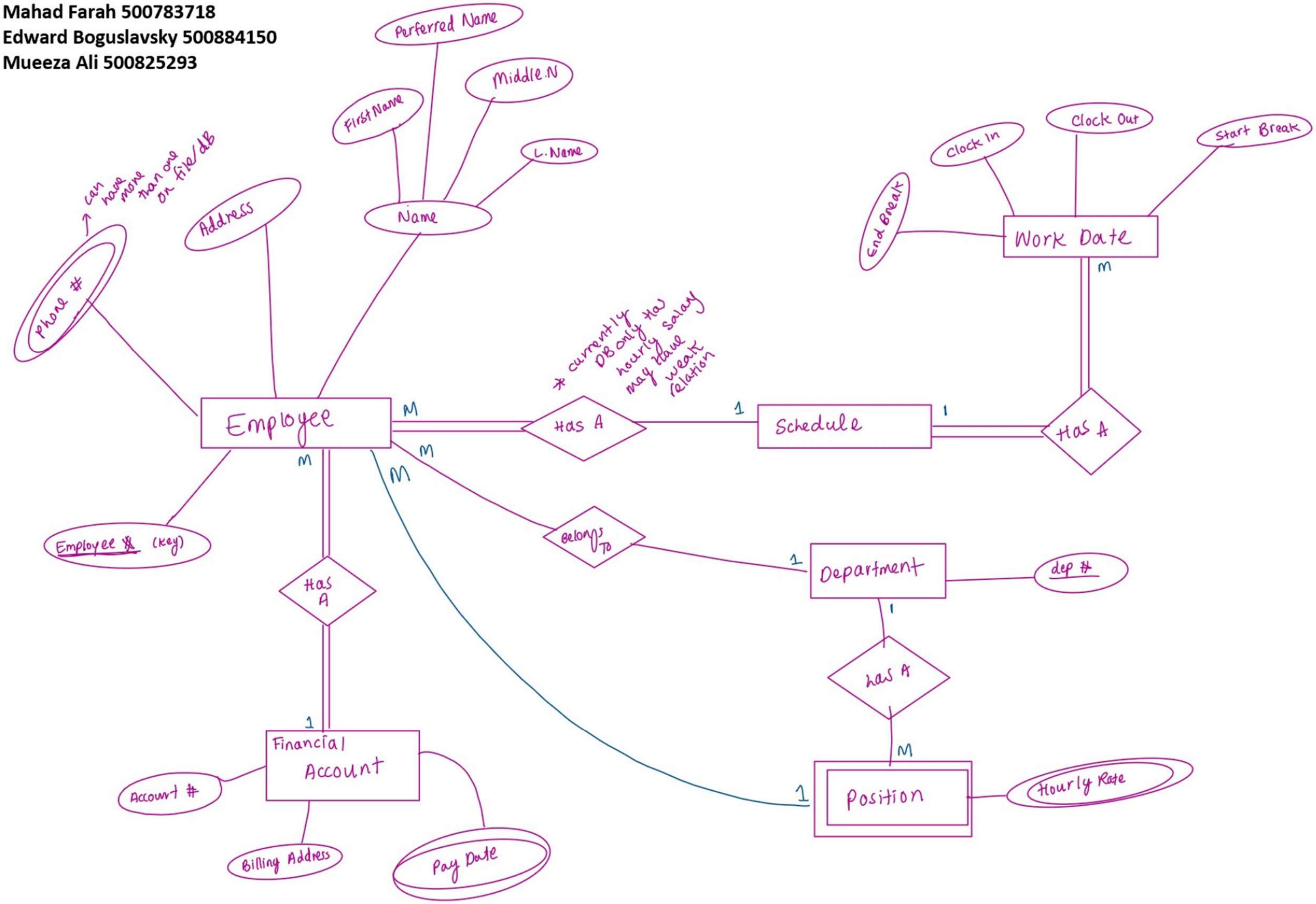
dep	Position	Hourly Rate
.	.	.
.		.
.	'	'
.	'	
.		

time off

vacation	Time Used	Remaining

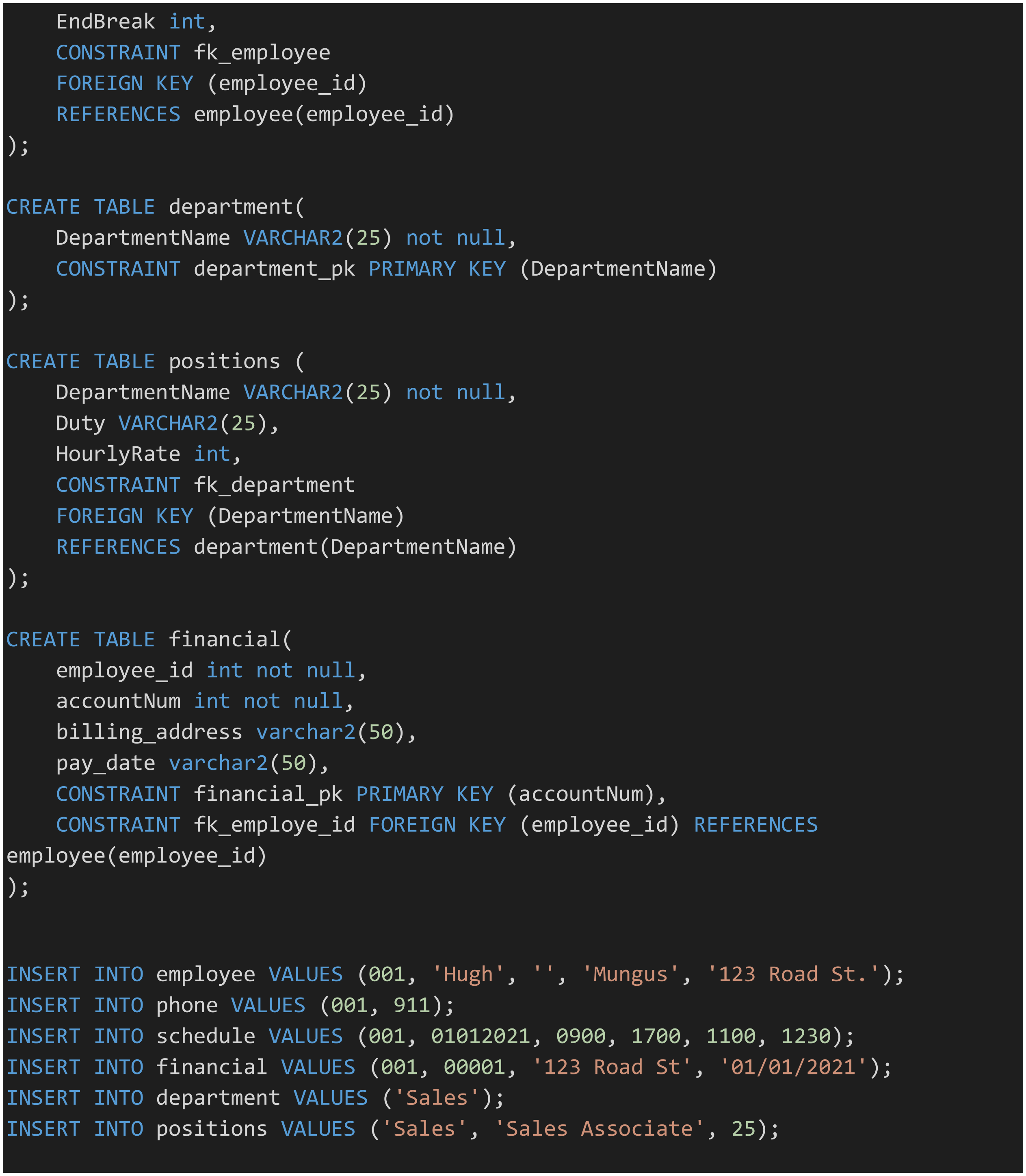
Assignment 2: ER Diagram

Mahad Farah 500783718
Edward Boguslavsky 500884150
Mueeza Ali 500825293



Assignment 3: Create Tables

```
#Mahad Farah 500783718
#Edward Boguslavsky 500884150
#Mueeza Ali 500825293

CREATE TABLE employee(
    employee_id int not null,
    first_name varchar2(50),
    middle_name varchar2(50),
    last_name varchar2(50),
    address varchar2(150),
    CONSTRAINT employee_pk PRIMARY KEY (employee_id)
);

CREATE TABLE phone(
    employee_id int not null,
    phone_number int,
    CONSTRAINT fk_phoneNumber
    FOREIGN KEY (employee_id)
    REFERENCES employee(employee_id)
);

CREATE TABLE schedule(
    employee_id int not null,
    DayWorked int,
    ClockIn int,
    ClockOut int,
    StartBreak int,
    EndBreak int,
    CONSTRAINT fk_employee
    FOREIGN KEY (employee_id)
    REFERENCES employee(employee_id)
);



CREATE TABLE schedule(
    employee_id int not null,
    DayWorked int,
    ClockIn int,
    ClockOut int,
    StartBreak int,
```



```
    EndBreak int,
    CONSTRAINT fk_employee
    FOREIGN KEY (employee_id)
    REFERENCES employee(employee_id)
);

CREATE TABLE department(
    DepartmentName VARCHAR2(25) not null,
    CONSTRAINT department_pk PRIMARY KEY (DepartmentName)
);

CREATE TABLE positions (
    DepartmentName VARCHAR2(25) not null,
    Duty VARCHAR2(25),
    HourlyRate int,
    CONSTRAINT fk_department
    FOREIGN KEY (DepartmentName)
    REFERENCES department(DepartmentName)
);

CREATE TABLE financial(
    employee_id int not null,
    accountNum int not null,
    billing_address varchar2(50),
    pay_date varchar2(50),
    CONSTRAINT financial_pk PRIMARY KEY (accountNum),
    CONSTRAINT fk_employe_id FOREIGN KEY (employee_id) REFERENCES
employee(employee_id)
);


INSERT INTO employee VALUES (001, 'Hugh', '', 'Mungus', '123 Road St.');
INSERT INTO phone VALUES (001, 911);
INSERT INTO schedule VALUES (001, 01012021, 0900, 1700, 1100, 1230);
INSERT INTO financial VALUES (001, 00001, '123 Road St', '01/01/2021');
INSERT INTO department VALUES ('Sales');
INSERT INTO positions VALUES ('Sales', 'Sales Associate', 25);
```

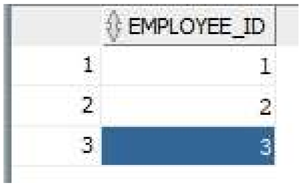
Assignment 4:

Insert Statements and Query Screenshots

Please note table creation is shown in assignment 3

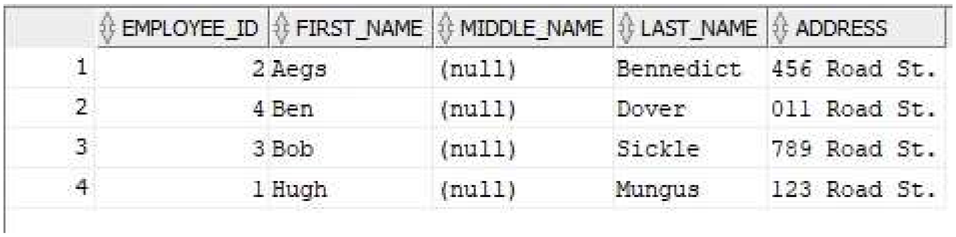
```
INSERT INTO employee VALUES (001, 'Hugh', '', 'Mungus', '123 Road St.');
INSERT INTO employee VALUES (002, 'Aegs', '', 'Bennedict', '456 Road St.');
INSERT INTO employee VALUES (003, 'Bob', '', 'Sickle', '789 Road St.');
INSERT INTO employee VALUES (004, 'Ben', '', 'Dover', '011 Road St.');

INSERT INTO phone VALUES (001, '911');
INSERT INTO phone VALUES (002, 922);
INSERT INTO phone VALUES (002, 9222);
INSERT INTO phone VALUES (002, 92222);
INSERT INTO phone VALUES (003, 933);

INSERT INTO schedule VALUES (001, '01012021', 0900, 1700, 1100, 1230);
INSERT INTO schedule VALUES (001, '01022021', 0910, 1710, 1100, 1230);
INSERT INTO schedule VALUES (002, '01022021', 0900, 1700, 1000, 1130); INSERT
INTO schedule VALUES (003, '01022021', 0855, 1705, 1230, 1400);

INSERT INTO department VALUES ('Sales');
INSERT INTO department VALUES ('IT');
INSERT INTO department VALUES ('Management');

INSERT INTO financial VALUES (001, 00001, '123 Road St', '01/01/2021');
INSERT INTO financial VALUES (002, 00002, '7 Franklin Blvd', '01/01/2021');
INSERT INTO financial VALUES (003, 00003, '86 Twenty First St', '01/01/2021');
INSERT INTO financial VALUES (004, 00004, '1127 Main St', '01/01/2021');
INSERT INTO positions VALUES ('Sales', 'Sales Associate', 25);

INSERT INTO positions VALUES ('Sales', 'Sales Associate', 16);

INSERT INTO positions VALUES ('IT', 'Service Technician', 24);

INSERT INTO positions VALUES ('Management', 'Assistant Manager', 26);

INSERT INTO positions VALUES ('Management', 'Head Manager', 34);
```

```
SELECT DISTINCT employee_id  FROM phone;/* This will be used to ensure all employees have a number
under the business note 004 did not have a number */
```



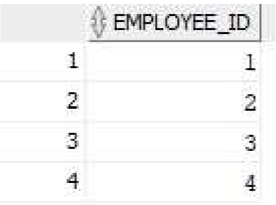
	EMPLOYEE_ID
1	1
2	2
3	3

```
SELECT * FROM employee ORDER BY first name ASC; /* This will be used to pull a master list of all
employees ( attendance on a trip etc ) / (secret santa) */
```



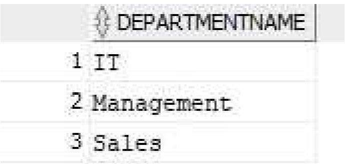
	EMPLOYEE_ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	ADDRESS
1	2	Aegs	(null)	Bennedict	456 Road St.
2	4	Ben	(null)	Dover	011 Road St.
3	3	Bob	(null)	Sickle	789 Road St.
4	1	Hugh	(null)	Mungus	123 Road St.

```
SELECT address FROM employee;/* Send employees their work from home laptop pull master list of
addresses*/
```



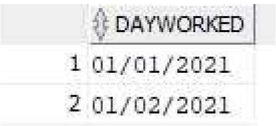
	EMPLOYEE_ID
1	1
2	2
3	3
4	4

```
SELECT DepartmentName FROM department; /*List all departements that are in the company*/
```



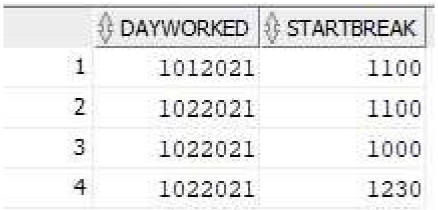
	DEPARTMENTNAME
1	IT
2	Management
3	Sales

```
SELECT DISTINCT DayWorked FROM schedule; /*Output what days the store open*/
```



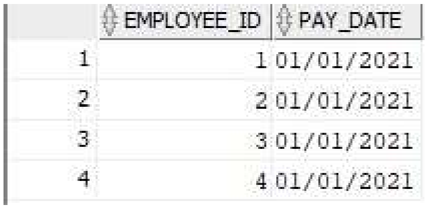
	DAYWORKED
1	01/01/2021
2	01/02/2021

```
SELECT DayWorked, StartBreak FROM schedule;/*List all break times on every day worked, in order to stagger breaks*/
```



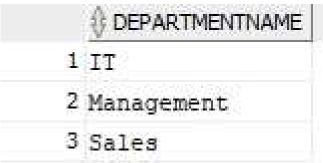
	DAYWORKED	STARTBREAK
1	1012021	1100
2	1022021	1100
3	1022021	1000
4	1022021	1230

```
SELECT employee_id, pay_date FROM financial ORDER BY employee_id ASC, pay_date ASC; /*List in ascending order who has been paid the most recently*/
```



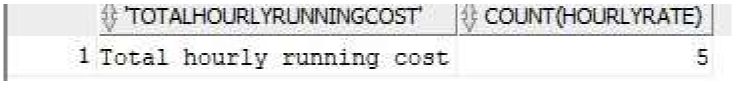
	EMPLOYEE_ID	PAY_DATE
1	1	01/01/2021
2	2	01/01/2021
3	3	01/01/2021
4	4	01/01/2021

```
SELECT DepartmentName FROM positions GROUP BY DepartmentName; /*List all departments at the tech retail store*/
```



	DEPARTMENTNAME
1	IT
2	Management
3	Sales

```
SELECT 'Total hourly running cost', COUNT(HourlyRate) FROM positions; /*Show the sum of all workers' hourly wages*/
```



	'TOTALHOURLYRUNNINGCOST'	COUNT(HOURLYRATE)
1	Total hourly running cost	5

Assignment 5: Advanced Addition

Insert Statements and Query Screenshots

Please note table creation is shown in assignment 3

```
INSERT INTO employee VALUES (001, 'Hugh', '', 'Mungus', '123 Road St.');

INSERT INTO employee VALUES (002, 'Aegs', '', 'Bennedict', '456 Road St.');

INSERT INTO employee VALUES (003, 'Bob', '', 'Sickle', '789 Road St.');

INSERT INTO employee VALUES (004, 'Ben', '', 'Dover', '011 Road St.');


INSERT INTO phone VALUES (001, '911');

INSERT INTO phone VALUES (002, 922);

INSERT INTO phone VALUES (002, 9222);

INSERT INTO phone VALUES (002, 92222);

INSERT INTO phone VALUES (003, 933);


INSERT INTO schedule VALUES (001, '01012021', 0900, 1700, 1100, 1230);

INSERT INTO schedule VALUES (001, '01022021', 0910, 1710, 1100, 1230);

INSERT INTO schedule VALUES (002, '01022021', 0900, 1700, 1000, 1130); INSERT

INTO schedule VALUES (003, '01022021', 0855, 1705, 1230, 1400);


INSERT INTO department VALUES ('Sales');

INSERT INTO department VALUES ('IT');

INSERT INTO department VALUES ('Management');


INSERT INTO financial VALUES (001, 00001, '123 Road St', '01/01/2021');

INSERT INTO financial VALUES (002, 00002, '7 Franklin Blvd', '01/01/2021');
```

INSERT INTO financial VALUES (003, 00003, '86 Twenty First St', '01/01/2021');

INSERT INTO financial VALUES (004, 00004, '1127 Main St', '01/01/2021');

INSERT INTO positions VALUES ('Sales', 'Sales Associate', 25);

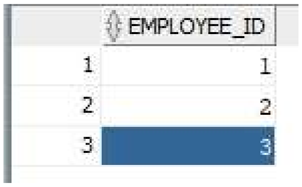
INSERT INTO positions VALUES ('Sales', 'Sales Associate', 16);

INSERT INTO positions VALUES ('IT', 'Service Technician', 24);

INSERT INTO positions VALUES ('Management', 'Assistant Manager', 26);

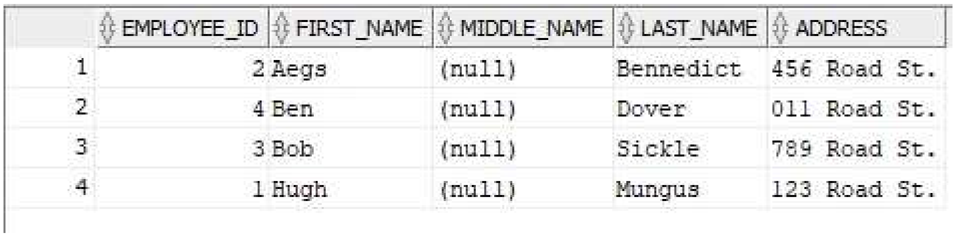
INSERT INTO positions VALUES ('Management', 'Head Manager', 34);

SELECT DISTINCT employee_id FROM phone;/* This will be used to ensure all employees have a number under the business note 004 did not have a number */



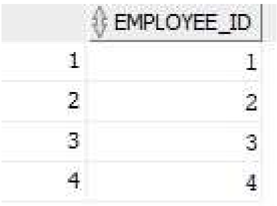
	EMPLOYEE_ID
1	1
2	2
3	3

SELECT * FROM employee ORDER BY first name ASC; /* This will be used to pull a master list of all employees (attendance on a trip etc) / (secret santa) */



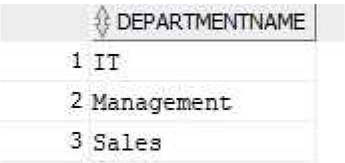
	EMPLOYEE_ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	ADDRESS
1	2	Aegs	(null)	Bennedict	456 Road St.
2	4	Ben	(null)	Dover	011 Road St.
3	3	Bob	(null)	Sickle	789 Road St.
4	1	Hugh	(null)	Mungus	123 Road St.

SELECT address FROM employee;/* Send employees their work from home laptop pull master list of addresses*/



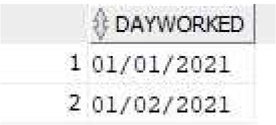
	EMPLOYEE_ID
1	1
2	2
3	3
4	4

SELECT DepartmentName FROM department; /*List all departements that are in the company*/



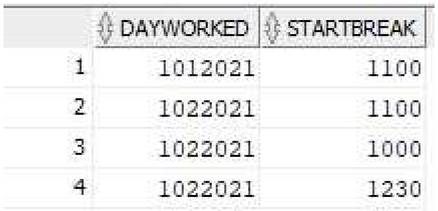
	DEPARTMENTNAME
1	IT
2	Management
3	Sales

SELECT DISTINCT DayWorked FROM schedule; /*Output what days the store open*/



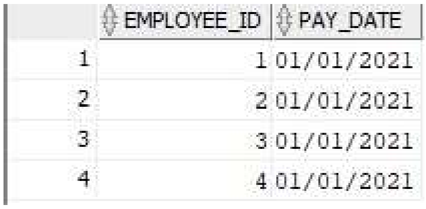
	DAYWORKED
1	01/01/2021
2	01/02/2021

SELECT DayWorked, StartBreak FROM schedule;/*List all break times on every day worked, in order to stagger breaks*/



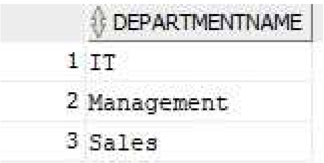
	DAYWORKED	STARTBREAK
1	1012021	1100
2	1022021	1100
3	1022021	1000
4	1022021	1230

SELECT employee_id, pay_date FROM financial ORDER BY employee_id ASC, pay_date ASC; /*List in ascending order who has been paid the most recently*/



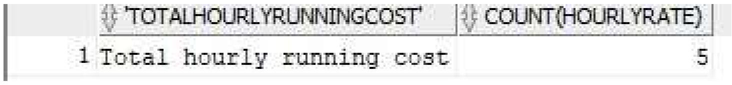
	EMPLOYEE_ID	PAY_DATE
1	1	01/01/2021
2	2	01/01/2021
3	3	01/01/2021
4	4	01/01/2021

SELECT DepartmentName FROM positions GROUP BY DepartmentName; /*List all departments at the tech retail store*/



	DEPARTMENTNAME
1	IT
2	Management
3	Sales

SELECT 'Total hourly running cost', COUNT(HourlyRate) FROM positions; /*Show the sum of all workers' hourly wages*/

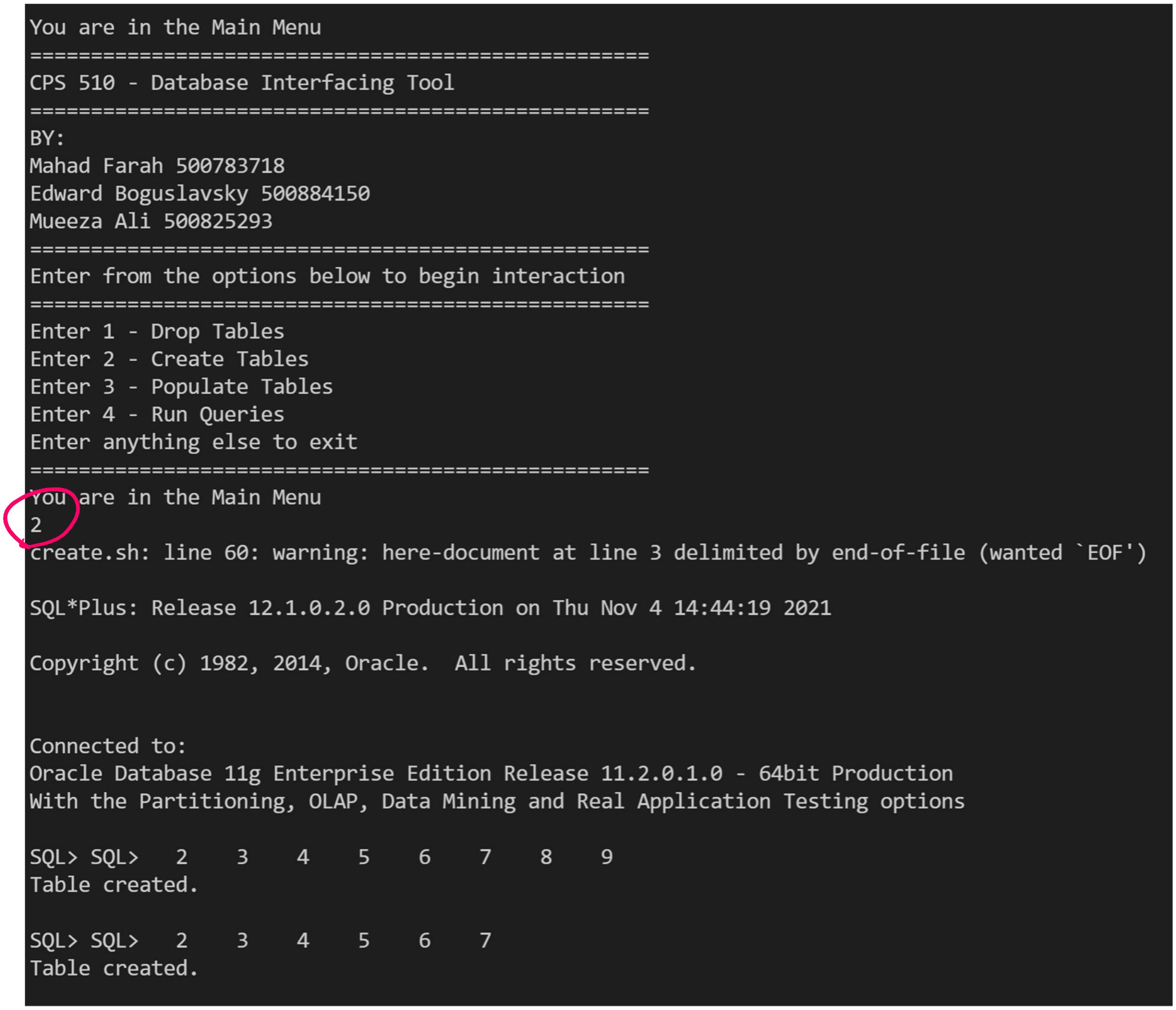


	'TOTALHOURLYRUNNINGCOST'	COUNT(HOURLYRATE)
1	Total hourly running cost	5

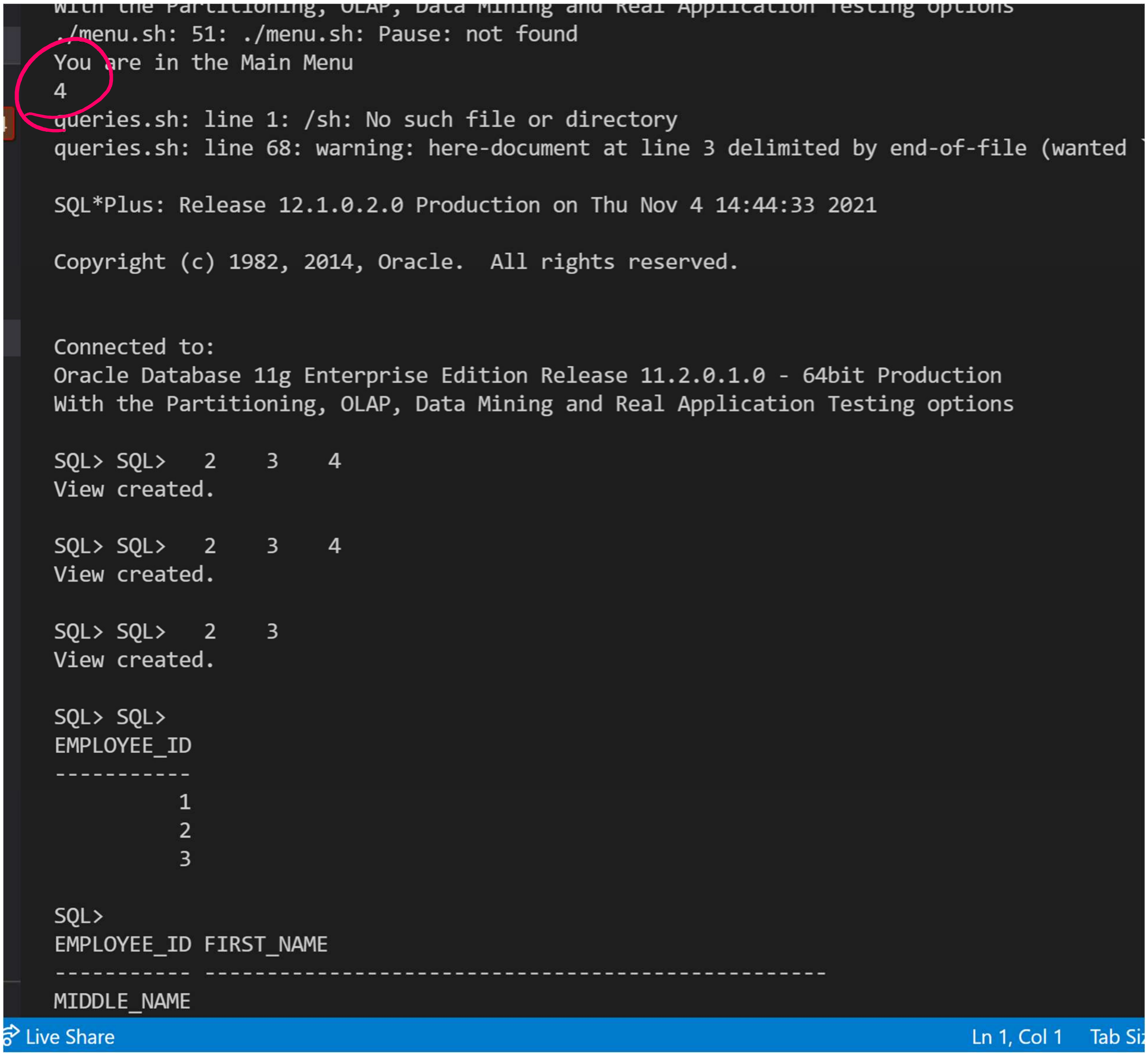
Assignment 5 – Main Menu Screenshots

A5 – CPS510:

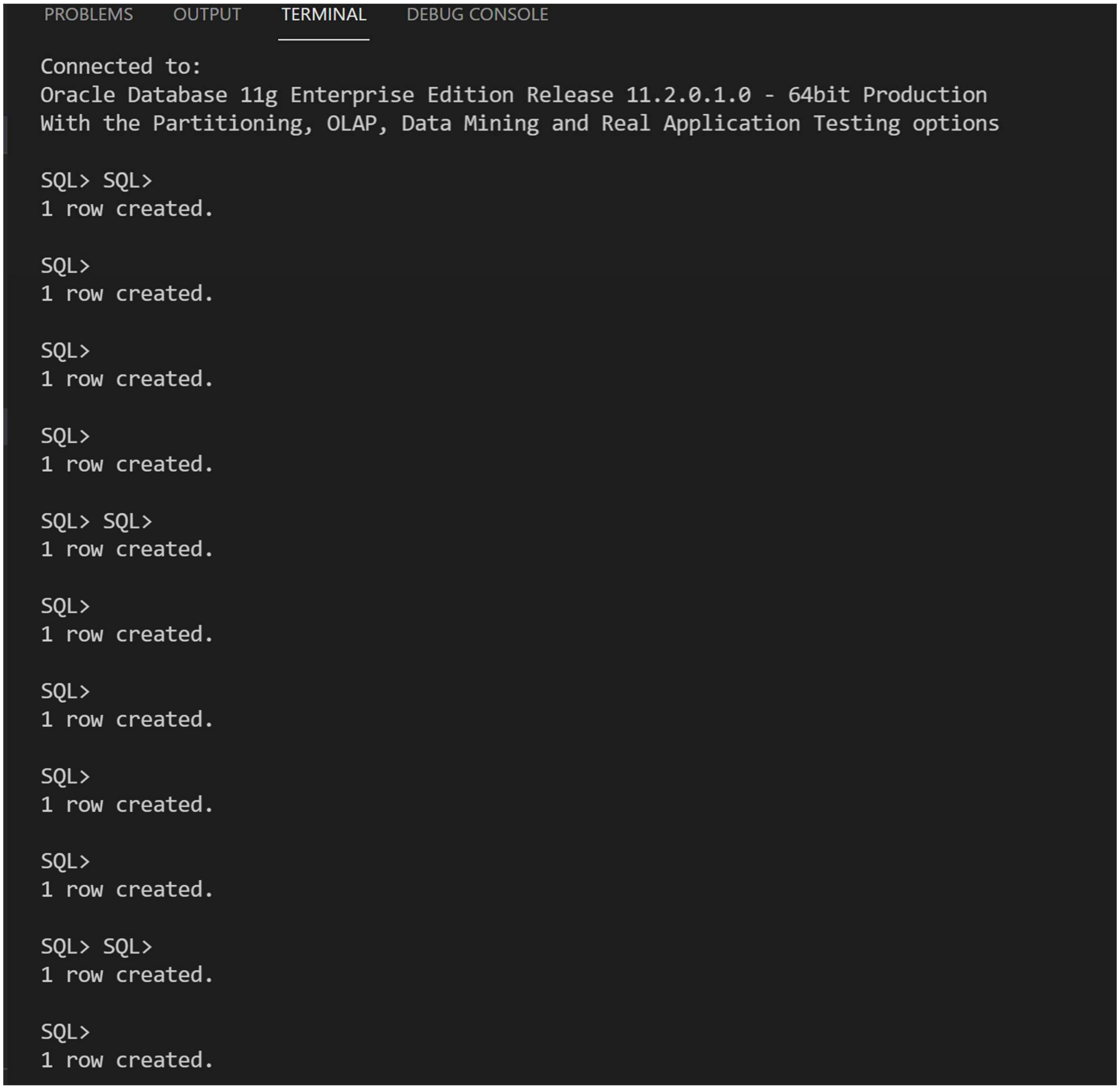
Below are screenshots that can be observed as was in the lab:



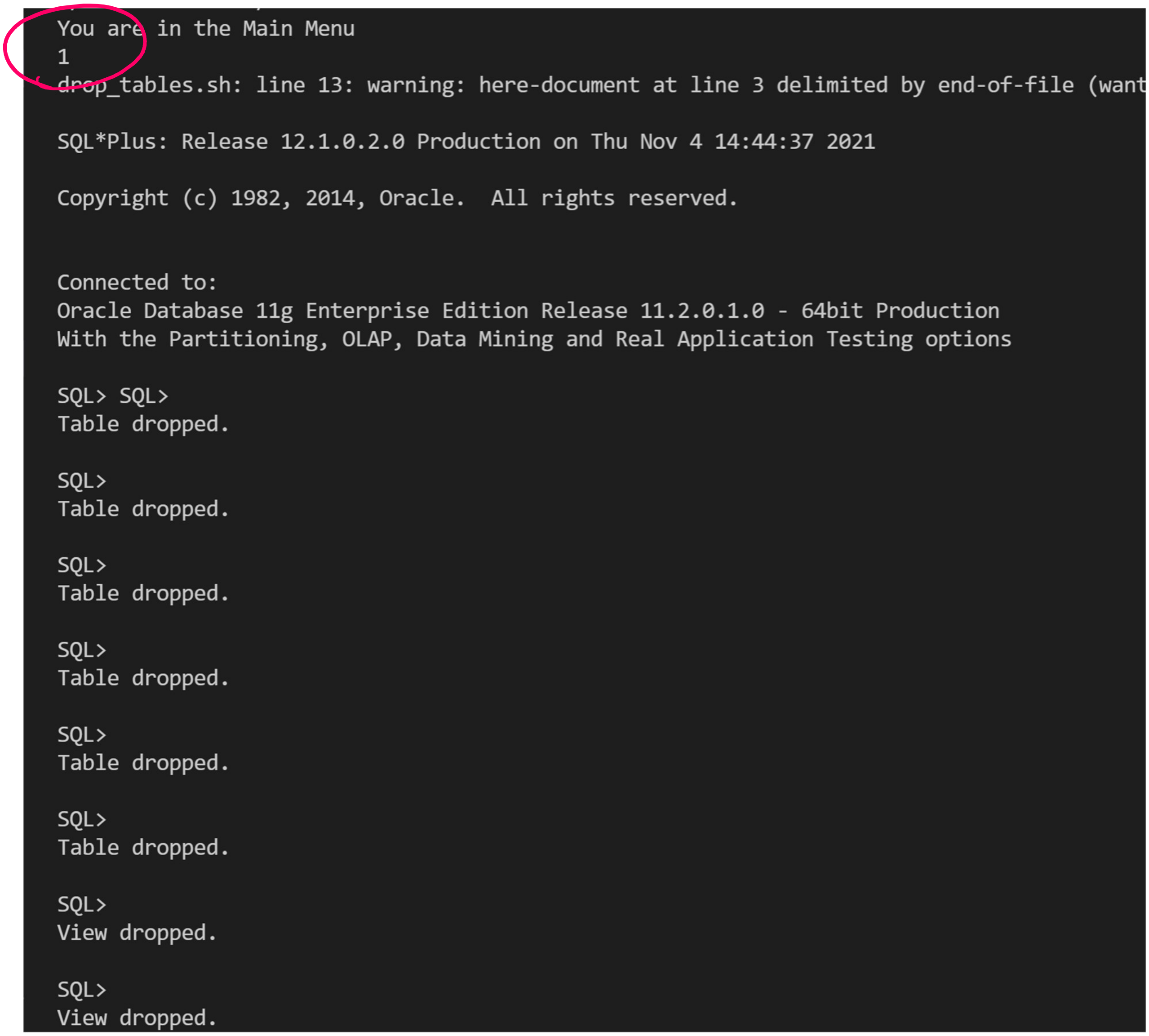
```
You are in the Main Menu
===================================================
CPS 510 - Database Interfacing Tool
===================================================
BY:
Mahad Farah 500783718
Edward Boguslavsky 500884150
Mueeza Ali 500825293
===================================================
Enter from the options below to begin interaction
===================================================
Enter 1 - Drop Tables
Enter 2 - Create Tables
Enter 3 - Populate Tables
Enter 4 - Run Queries
Enter anything else to exit
===================================================
You are in the Main Menu
2
create.sh: line 60: warning: here-document at line 3 delimited by end-of-file (wanted `EOF')

SQL*Plus: Release 12.1.0.2.0 Production on Thu Nov 4 14:44:19 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>   2    3    4    5    6    7    8    9
Table created.

SQL> SQL>   2    3    4    5    6    7
Table created.
```



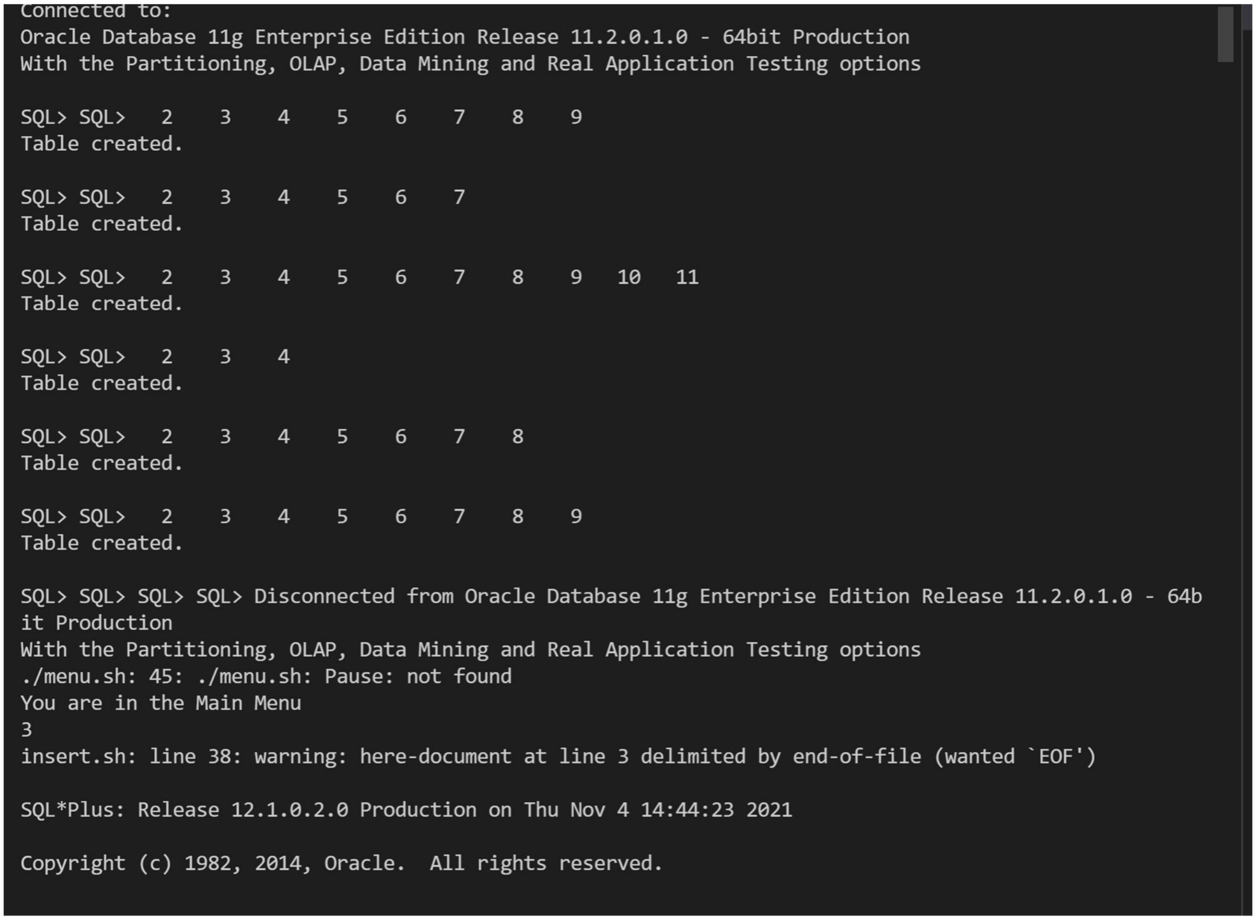
```
With the Partitioning, OLAP, Data Mining and Real Application Testing options
./menu.sh: 51: ./menu.sh: Pause: not found
You are in the Main Menu
4
queries.sh: line 1: /sh: No such file or directory
queries.sh: line 68: warning: here-document at line 3 delimited by end-of-file (wanted `

SQL*Plus: Release 12.1.0.2.0 Production on Thu Nov 4 14:44:33 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>   2    3    4
View created.

SQL> SQL>   2    3    4
View created.

SQL> SQL>   2    3
View created.

SQL> SQL>
EMPLOYEE_ID
-----------
          1
          2
          3

SQL>
EMPLOYEE_ID FIRST_NAME
----------- --------------------------------------------------
MIDDLE_NAME
```



```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.
```



```
You are in the Main Menu
1
drop_tables.sh: line 13: warning: here-document at line 3 delimited by end-of-file (want

SQL*Plus: Release 12.1.0.2.0 Production on Thu Nov 4 14:44:37 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
View dropped.

SQL>
View dropped.
```



```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>   2    3    4    5    6    7    8    9
Table created.

SQL> SQL>   2    3    4    5    6    7
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10   11
Table created.

SQL> SQL>   2    3    4
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9
Table created.

SQL> SQL> SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64b
it Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
./menu.sh: 45: ./menu.sh: Pause: not found
You are in the Main Menu
3
insert.sh: line 38: warning: here-document at line 3 delimited by end-of-file (wanted `EOF')

SQL*Plus: Release 12.1.0.2.0 Production on Thu Nov 4 14:44:23 2021

Copyright (c) 1982, 2014, Oracle.  All rights reserved.
```

Assignment 6:

Mahad Farah	500783718
Edward Boguslavsky	500884150
Mueeza Ali	500825293

Below are the listed Function Dependencies. For SQL code please refer to a5

Employee Table *(Table 1)*

- employee(employee_id, first_name, middle_name ,last_name, address, store_city)
- employee_id -> first_name
- employee_id -> middle_name
- employee_id -> last_name
- employee_id -> address
- employee_id -> store_city

Phone Table *(Table 2)*

- employee_id -> phone_number

Schedule Table *(Table 3)*

- employee_id -> DayWorked
- employee_id -> ClockIn
- employee_id -> ClockOut
- employee_id -> StartBreak
- employee_id -> EndBreak

Department Table *(Table 4)*

No Dependencies

Positions Table *(Table 5)*

- department_name -> duty

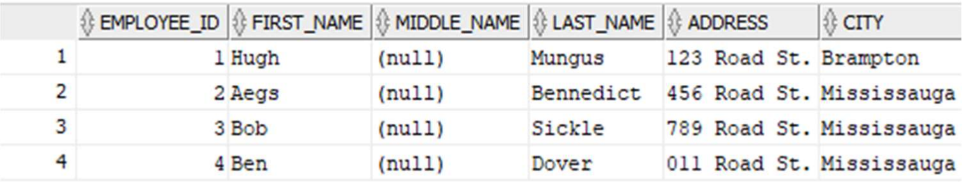
Financial Table *(Table 6)*

- financial(accountNum, billing_address, store_city, pay_date)
- accountNum -> billing_address
- accountNum -> store_city
- accountNum -> pay_date

Assignment 7: Continuation

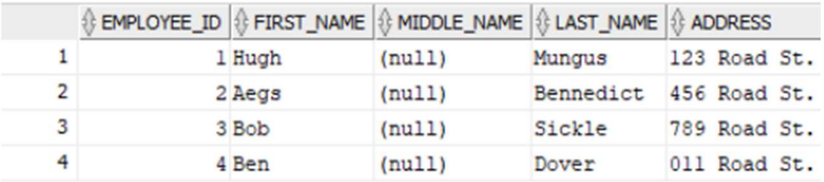
Employee Table *(Table 1)*

- employee(employee_id, first_name, middle_name ,last_name, address, city)
- employee_id -> first_name
- employee_id -> middle_name
- employee_id -> last_name
- employee_id -> address
- employee_id -> city

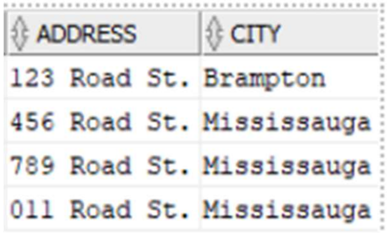


	EMPLOYEE_ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	ADDRESS	CITY
1	1	Hugh	(null)	Mungus	123 Road St.	Brampton
2	2	Aegs	(null)	Bennedict	456 Road St.	Mississauga
3	3	Bob	(null)	Sickle	789 Road St.	Mississauga
4	4	Ben	(null)	Dover	011 Road St.	Mississauga

The employee table above has the transitive dependency as follows: employee_id -> address -> city. Converting to 2NF results in the tables below.



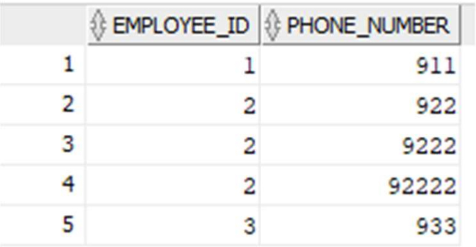
	EMPLOYEE_ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	ADDRESS
1	1	Hugh	(null)	Mungus	123 Road St.
2	2	Aegs	(null)	Bennedict	456 Road St.
3	3	Bob	(null)	Sickle	789 Road St.
4	4	Ben	(null)	Dover	011 Road St.



ADDRESS	CITY
123 Road St.	Brampton
456 Road St.	Mississauga
789 Road St.	Mississauga
011 Road St.	Mississauga

Phone Table *(Table 2)*

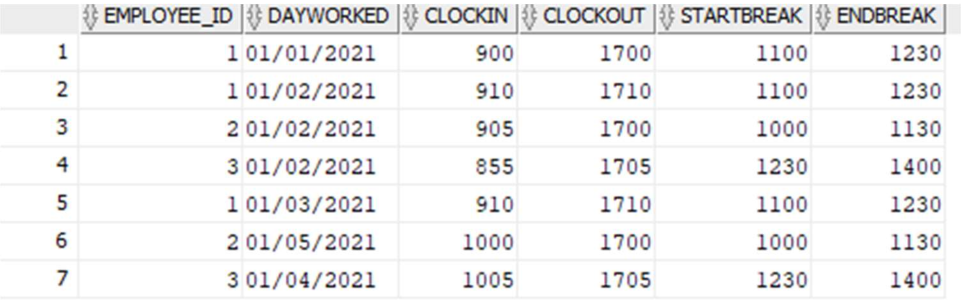
- employee_id -> phone_number



	EMPLOYEE_ID	PHONE_NUMBER
1	1	911
2	2	922
3	2	9222
4	2	92222
5	3	933

Schedule Table *(Table 3)*

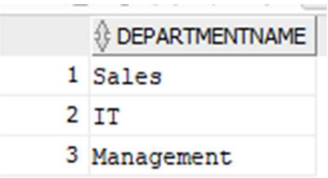
- employee_id -> DayWorked
- employee_id -> ClockIn
- employee_id -> ClockOut
- employee_id -> StartBreak
- employee_id -> EndBreak



	EMPLOYEE_ID	DAYWORKED	CLOCKIN	CLOCKOUT	STARTBREAK	ENDBREAK
1	1	01/01/2021	900	1700	1100	1230
2	1	01/02/2021	910	1710	1100	1230
3	2	01/02/2021	905	1700	1000	1130
4	3	01/02/2021	855	1705	1230	1400
5	1	01/03/2021	910	1710	1100	1230
6	2	01/05/2021	1000	1700	1000	1130
7	3	01/04/2021	1005	1705	1230	1400

Department Table *(Table 4)*

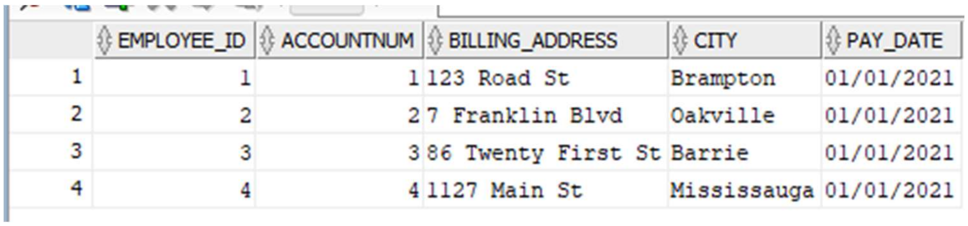
No Dependencies



	DEPARTMENTNAME
1	Sales
2	IT
3	Management

Financial Table *(Table 6)*

- financial(accountNum, billing_address, store_city, pay_date)
- accountNum -> billing_address
- accountNum -> store_city
- accountNum -> pay_date



	EMPLOYEE_ID	ACCOUNTNUM	BILLING_ADDRESS	CITY	PAY_DATE
1	1	1	123 Road St	Brampton	01/01/2021
2	2	2	7 Franklin Blvd	Oakville	01/01/2021
3	3	3	86 Twenty First St	Barrie	01/01/2021
4	4	4	1127 Main St	Mississauga	01/01/2021

Selected Table of Focus for A7:

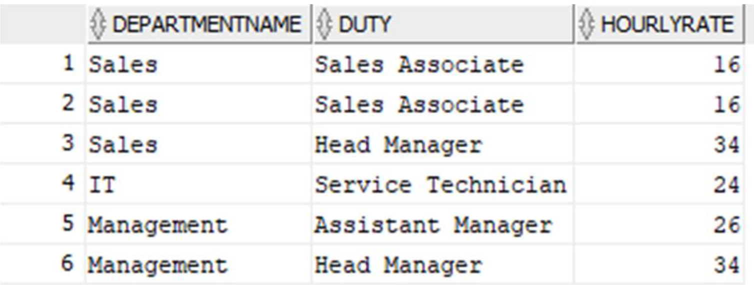
Positions Table *(Table 5 Reverted to Not 3NF form)*

Compound FD’s to 2NF

Department(DepartmentName, Duty, HourlyRate)

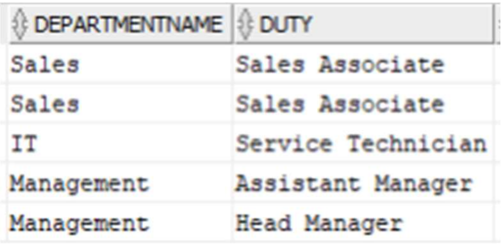
Canidate_key {DepartmentName, Duty}

-> Note that DeparmentName and Duty are compound attributes which HourlyRate is dependent on. Note this is in 1NF as we have a compound + partial dependency.



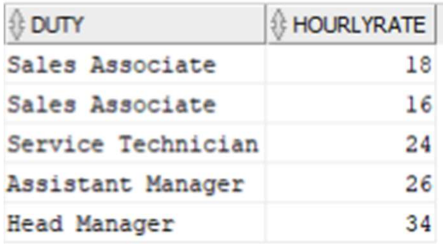
	DEPARTMENTNAME	DUTY	HOURLYRATE
1	Sales	Sales Associate	16
2	Sales	Sales Associate	16
3	Sales	Head Manager	34
4	IT	Service Technician	24
5	Management	Assistant Manager	26
6	Management	Head Manager	34

-> To convert the above relation to 2NF,
we need to split the table into two tables such as :
Table 1: DeparmentName, Duty
Table 2: Duty, Hourly Rate



DEPARTMENTNAME	DUTY
Sales	Sales Associate
Sales	Sales Associate
IT	Service Technician
Management	Assistant Manager
Management	Head Manager

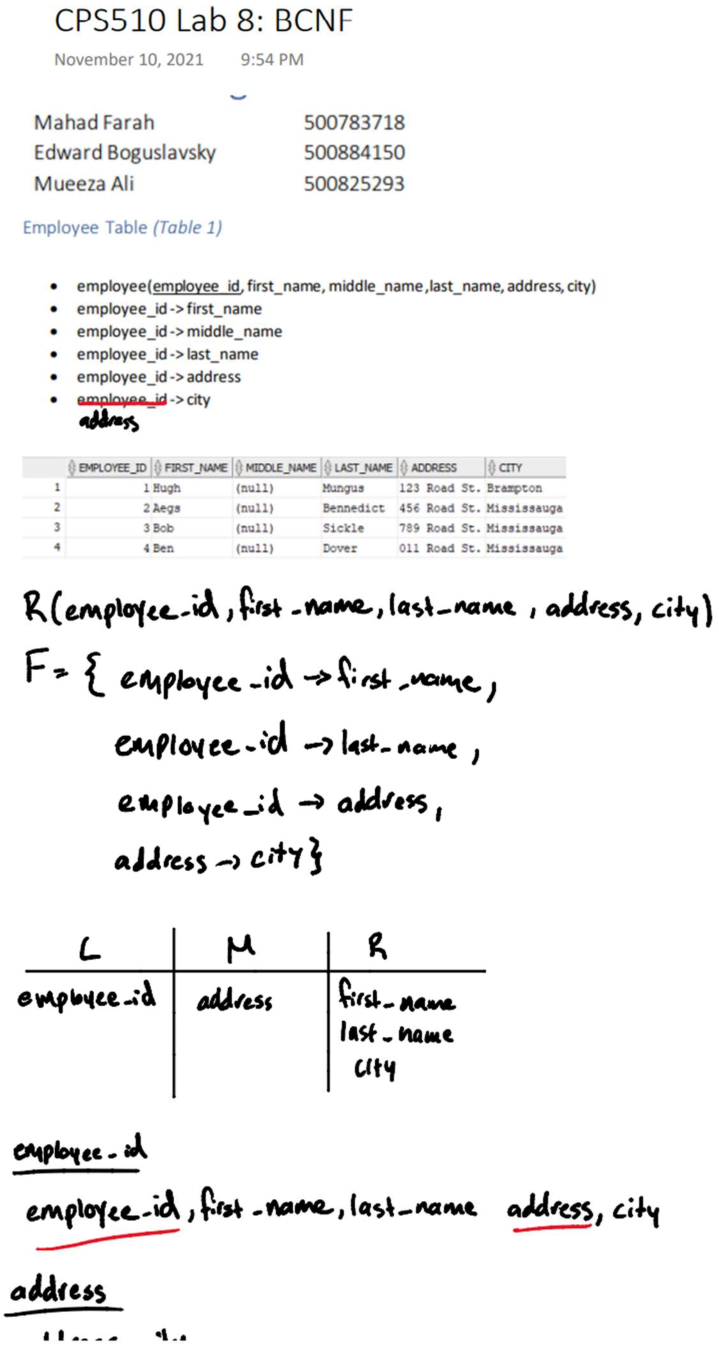
DepartmentName -> Duty



DUTY	HOURLYRATE
Sales Associate	18
Sales Associate	16
Service Technician	24
Assistant Manager	26
Head Manager	34

Duty -> HourlyRate

Assignment 8



CPS510 Lab 8: BCNF

November 10, 2021 9:54 PM

Mahad Farah	500783718
Edward Boguslavsky	500884150
Mueeza Ali	500825293

Employee Table *(Table 1)*

- employee(employee_id, first_name, middle_name, last_name, address, city)
- employee_id -> first_name
- employee_id -> middle_name
- employee_id -> last_name
- employee_id -> address
- ~~employee_id~~ -> city
  address

	EMPLOYEE_ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	ADDRESS	CITY
1	1	Hugh	(null)	Mungus	123 Road St.	Brampton
2	2	Aegs	(null)	Bennedict	456 Road St.	Mississauga
3	3	Bob	(null)	Sickle	789 Road St.	Mississauga
4	4	Ben	(null)	Dover	011 Road St.	Mississauga

R(employee-id, first-name, last-name, address, city)

F = { employee-id -> first-name,
employee-id -> last-name,
employee-id -> address,
address -> city }

L	M	R
employee-id	address	first-name last-name city

employee-id

employee-id, first-name, last-name address, city

address

address

address, city

∴, the table is in 1NF as it violates 2NF.

ey: employee_id, address

R_1(employee_id, first_name)

R_2(employee_id, last_name)

R_3(employee_id, address)

R_4(address, city)

S(employee_id, first_name last_name)

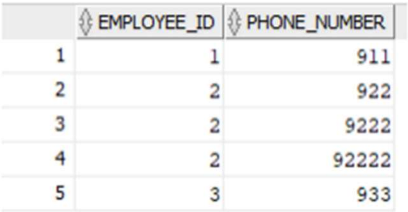
T(employee_id, address, city)

S, FD: (employee_id -> first_name,
employee_id -> last_name)

T, FD: (employee_id -> address,
employee_id -> city)

Phone Table *(Table 2)*

- employee_id -> phone_number

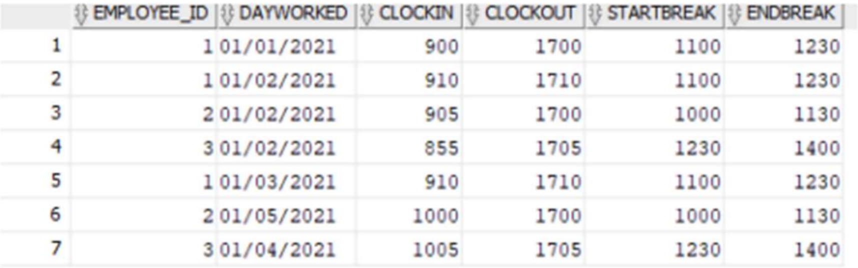


	EMPLOYEE_ID	PHONE_NUMBER
1	1	911
2	2	922
3	2	9222
4	2	92222
5	3	933

Schedule Table *(Table 3)*

- employee_id -> DayWorked
- employee_id -> ClockIn
- employee_id -> ClockOut

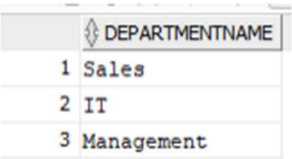
- employee_id -> StartBreak
- employee_id -> EndBreak



	EMPLOYEE_ID	DAYWORKED	CLOCKIN	CLOCKOUT	STARTBREAK	ENDBREAK
1	1	01/01/2021	900	1700	1100	1230
2	1	01/02/2021	910	1710	1100	1230
3	2	01/02/2021	905	1700	1000	1130
4	3	01/02/2021	855	1705	1230	1400
5	1	01/03/2021	910	1710	1100	1230
6	2	01/05/2021	1000	1700	1000	1130
7	3	01/04/2021	1005	1705	1230	1400

Department Table *(Table 4)*

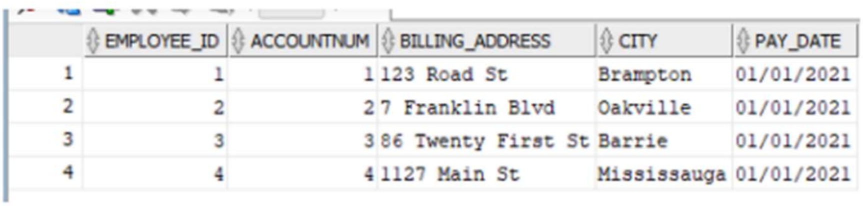
No Dependencies



	DEPARTMENTNAME
1	Sales
2	IT
3	Management

Financial Table *(Table 6)*

- financial(accountNum, billing_address, store_city, pay_date)
- accountNum -> billing_address
- accountNum -> store_city
- accountNum -> pay_date



	EMPLOYEE_ID	ACCOUNTNUM	BILLING_ADDRESS	CITY	PAY_DATE
1	1	1	123 Road St	Brampton	01/01/2021
2	2	2	7 Franklin Blvd	Oakville	01/01/2021
3	3	3	86 Twenty First St	Barrie	01/01/2021
4	4	4	1127 Main St	Mississauga	01/01/2021

Selected Table of Focus for A7:

Positions Table *(Table 5 Reverted to Not 3NF form)*

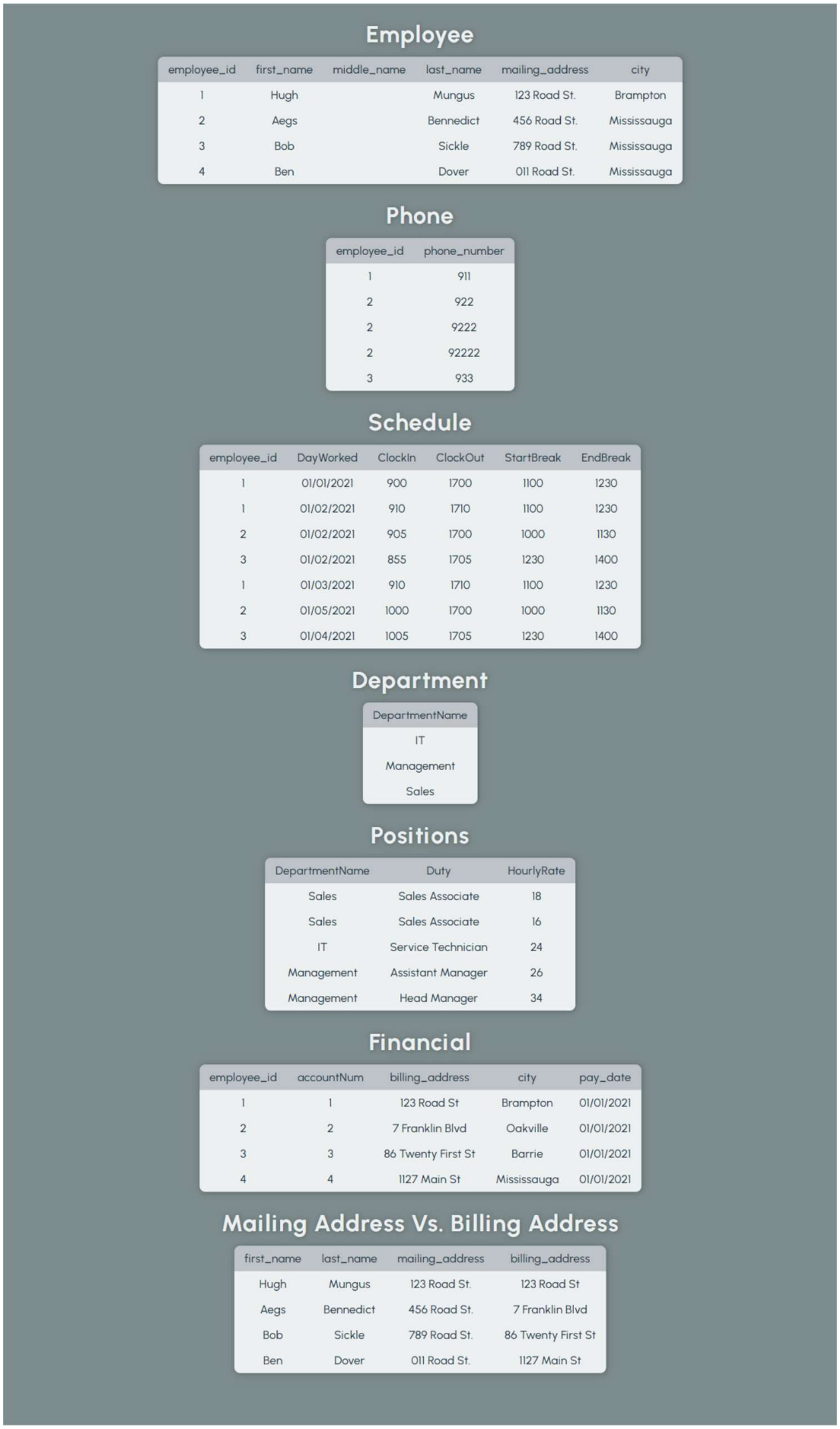
Compound FD's to 2NF

Department(DepartmentName, Duty, HourlyRate)

Canidate_key {DepartmentName, Duty}

-> Note that DeparmentName and Duty are compound attributes which HourlyRate is dependent on. Note this is in 1NF as we have a compound + partial dependency.

A9 GUI Screenshot:



Employee

employee_id	first_name	middle_name	last_name	mailing_address	city
1	Hugh		Mungus	123 Road St.	Brampton
2	Aegs		Bennedict	456 Road St.	Mississauga
3	Bob		Sickle	789 Road St.	Mississauga
4	Ben		Dover	011 Road St.	Mississauga

Phone

employee_id	phone_number
1	911
2	922
2	9222
2	92222
3	933

Schedule

employee_id	DayWorked	ClockIn	ClockOut	StartBreak	EndBreak
1	01/01/2021	900	1700	1100	1230
1	01/02/2021	910	1710	1100	1230
2	01/02/2021	905	1700	1000	1130
3	01/02/2021	855	1705	1230	1400
1	01/03/2021	910	1710	1100	1230
2	01/05/2021	1000	1700	1000	1130
3	01/04/2021	1005	1705	1230	1400

Department

DepartmentName
IT
Management
Sales

Positions

DepartmentName	Duty	HourlyRate
Sales	Sales Associate	18
Sales	Sales Associate	16
IT	Service Technician	24
Management	Assistant Manager	26
Management	Head Manager	34

Financial

employee_id	accountNum	billing_address	city	pay_date
1	1	123 Road St	Brampton	01/01/2021
2	2	7 Franklin Blvd	Oakville	01/01/2021
3	3	86 Twenty First St	Barrie	01/01/2021
4	4	1127 Main St	Mississauga	01/01/2021

Mailing Address Vs. Billing Address

first_name	last_name	mailing_address	billing_address
Hugh	Mungus	123 Road St.	123 Road St
Aegs	Bennedict	456 Road St.	7 Franklin Blvd
Bob	Sickle	789 Road St.	86 Twenty First St
Ben	Dover	011 Road St.	1127 Main St

Assignment 10

$\Pi_{employee\_id}$ (phone)

```
SELECT DISTINCT employee_id FROM phone;
```

Mueeza

σ_{first_name} (employee)

```
SELECT * FROM employee ORDER BY first_name ASC;
```

Mueeza

$\Pi_{employee\_id}$ (employee)

```
SELECT employee_id FROM employee;
```

Mueeza

$\Pi_{departmentName}$ (department)

```
SELECT DepartmentName FROM department; /*List all departments that are
in the company*/
```

Mueeza

$\Pi_{dayWorked}$ (schedule)

```
SELECT DISTINCT DayWorked FROM schedule; /*Output what days the store
open*/
```

Mueeza

$\Pi_{DayWorked,\ StartBreak}$ (schedule)

```
SELECT DayWorked, StartBreak FROM schedule; /*List all break times on
every day worked, in order to stagger breaks*/
```

Mueeza

$\Pi_{employee\_id,\ pay\_date}$ (financial)

```
SELECT employee_id, pay_date FROM financial ORDER BY employee_id ASC,
pay_date ASC;
SELECT DepartmentName FROM positions GROUP BY DepartmentName;
SELECT 'Total hourly running cost', COUNT(HourlyRate) FROM positions;
```

Edward

$\prod_{\text{employee\_id, first\_name, last\_name, phone\_number}} (\text{employee} \bowtie \text{phone})$

```sql
/* Assignent 4.5 */
SELECT employee.employee_id, employee.first_name, employee.last_name,
phone.phone_number /*List all employee phone numbers */
FROM employee
INNER JOIN phone ON phone.employee_id = employee.employee_id;
```

Edward

$\prod_{\text{first\_name, last\_name, address, billing\_address}} (\text{employee} \bowtie \text{financial})$

```sql
SELECT employee.first_name, employee.last_name, employee.address,
financial.billing_address
FROM employee
INNER JOIN financial ON employee.employee_id = financial.employee_id;
```

Edward

$\prod_{\text{first\_name, last\_name, DayWorked}} (\text{employee} \bowtie \text{schedule})$

```sql
SELECT employee.first_name, employee.last_name, schedule.DayWorked
/*List all days worked and the employee name */
FROM employee
INNER JOIN schedule ON employee.employee_id = schedule.employee_id;
```

Edward

$P_{\text{city}} (\text{employee}) \cup P_{\text{city}} (\text{financial})$

```sql
SELECT city FROM employee
UNION
SELECT city FROM financial
ORDER BY city;
```

Mahad

$\prod_{*} (\sigma_{\text{city <> 'Mississauga'}} (\text{employee}))$

```sql
SELECT * from employee
MINUS select * from employee
WHERE city <> 'Mississauga';
```

Mahad

$\prod_{\text{positions, HourlyRate}} (\sigma_{\text{HourlyRate > 25}} (\text{positions}))$

```sql
SELECT COUNT(HourlyRate)
FROM positions
WHERE HourlyRate > 25;
```

Mahad

$\Pi_{\text{HourlyRate}}\ (\sigma_{\text{Topic =”departmentName” and departmentName=”Sales”}}\ (\text{positions}))$

```
SELECT HourlyRate
FROM positions
WHERE EXISTS (SELECT DepartmentName FROM department WHERE
DepartmentName = 'Sales');
```

Mahad

${}_{\text{EMPLOYEESTARTTIMES, CLOCKIN}}\ G_{\text{COUNT (EMPLOYEESTARTTIMES)}}$

```
SELECT COUNT(employee_id) as EmployeeStartTimes, ClockIn
FROM schedule
GROUP BY ClockIn;
```

Mahad

${}_{\text{HourlyRate, Duty}}\ G_{\text{AVG (HourlyRate)}}$

```
SELECT AVG(HourlyRate), Duty
FROM positions
GROUP BY Duty
HAVING AVG(HourlyRate) < 20;
```

Mahad

Final Table Creation Code and Remarks:

Table Creation Code

```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
sqlplus64
"m262ali/####@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.
ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF

CREATE TABLE employee(
    employee_id int not null,
    first_name varchar2(50),
    middle_name varchar2(50),
    last_name varchar2(50),
    address varchar2(150),
    city varchar2(150),
    CONSTRAINT employee_pk PRIMARY KEY (employee_id)
);

CREATE TABLE phone(
    employee_id int not null,
    phone_number int,
    CONSTRAINT fk_phoneNumber
    FOREIGN KEY (employee_id)
    REFERENCES employee(employee_id)
);

CREATE TABLE schedule(
    employee_id int not null,
    DayWorked varchar2(50),
    ClockIn int,
    ClockOut int,
    StartBreak int,
    EndBreak int,
    CONSTRAINT fk_employee
    FOREIGN KEY (employee_id)
    REFERENCES employee(employee_id)
);

CREATE TABLE department(
    DepartmentName VARCHAR2(25) not null,
    CONSTRAINT department_pk PRIMARY KEY (DepartmentName)
);

CREATE TABLE positions (
    DepartmentName VARCHAR2(25) not null,
    Duty VARCHAR2(25),
    HourlyRate int,
    CONSTRAINT fk_department
    FOREIGN KEY (DepartmentName)
    REFERENCES department(DepartmentName)
);

CREATE TABLE belongTo(
    employee_id VARCHAR2(25) not null,
    Duty VARCHAR2(25),
    FOREIGN KEY (Duty)
    CONSTRAINT employee_pk PRIMARY KEY (employee_id)
);

CREATE TABLE financial(
    employee_id int not null,
    accountNum int not null,
    billing_address varchar2(50),
    city varchar2(150),
    pay_date varchar2(50),
    CONSTRAINT financial_pk PRIMARY KEY (accountNum),
    CONSTRAINT fk_employe_id FOREIGN KEY (employee_id) REFERENCES
employee(employee_id)
);
```

Instructions:

```
To run the GUI, follow these steps:

   1)   Run OpenVPN
   2)   Go to "https://webdev.scs.ryerson.ca/~m24farah/510dbquery.php" in your browser
   3)   Enjoy
```