ITEC 4200 Advanced Database Semester Project

Instructor: Lissa Pollacia
Student Name: Quan Tran
Semester: FALL 2015
Name of Project: CONVENIENCE STORE
Email: qtran3@ggc.edu
Phone number(s): 770-910-3300

GEORGIA GWINNETT COLLEGE

Part I. Executive Summary
CONVENIENCE STORE DATABASE

This database contains all information to help manage a small scale business at a convenience store. It allows the manager to keep track of staffs works, keep tracking of selling products. It also allows customer to view products' information. It also keep track of changing process of product when the store hosts sale off events.

Every staff should have a profile so that the manager or customer can contact to if necessary. The system will keep track the works of staffs, as well as the salary paid to them.

Every product should have its information such as: name, origin, brand, expired date, the cost imported. The store will decide its selling price.

Selling price of product may change every time the store host events like sale-off-hour, or just follow the market price. We'd keep track prices; of every product. So that, when a customer buy a product, the system will find the latest price applying to that product.

Customer should have a profile, so that we can contact.

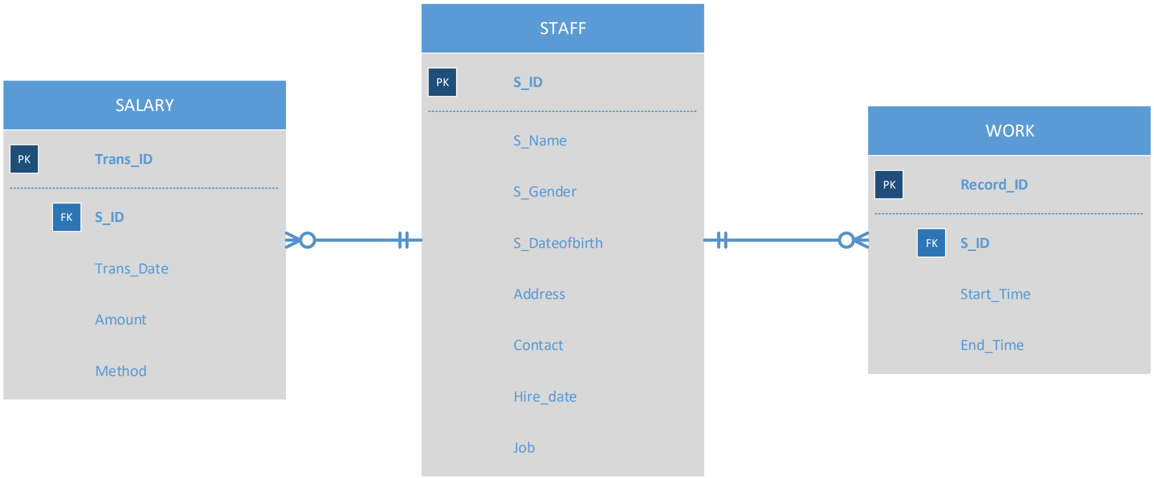
Orders made by customers may earn some discounts due to when the store host events, or just by "a lucky customer".

To simplify the use of the database, a switchboard is also included to make it simpler to view the tables, queries, forms and reports in a menu-driven format. The database will be implemented using Oracle 11g Express Database.

Part II: THE E-R DIAGRAM AND SCHEMA

Schema of the database will include these tables:

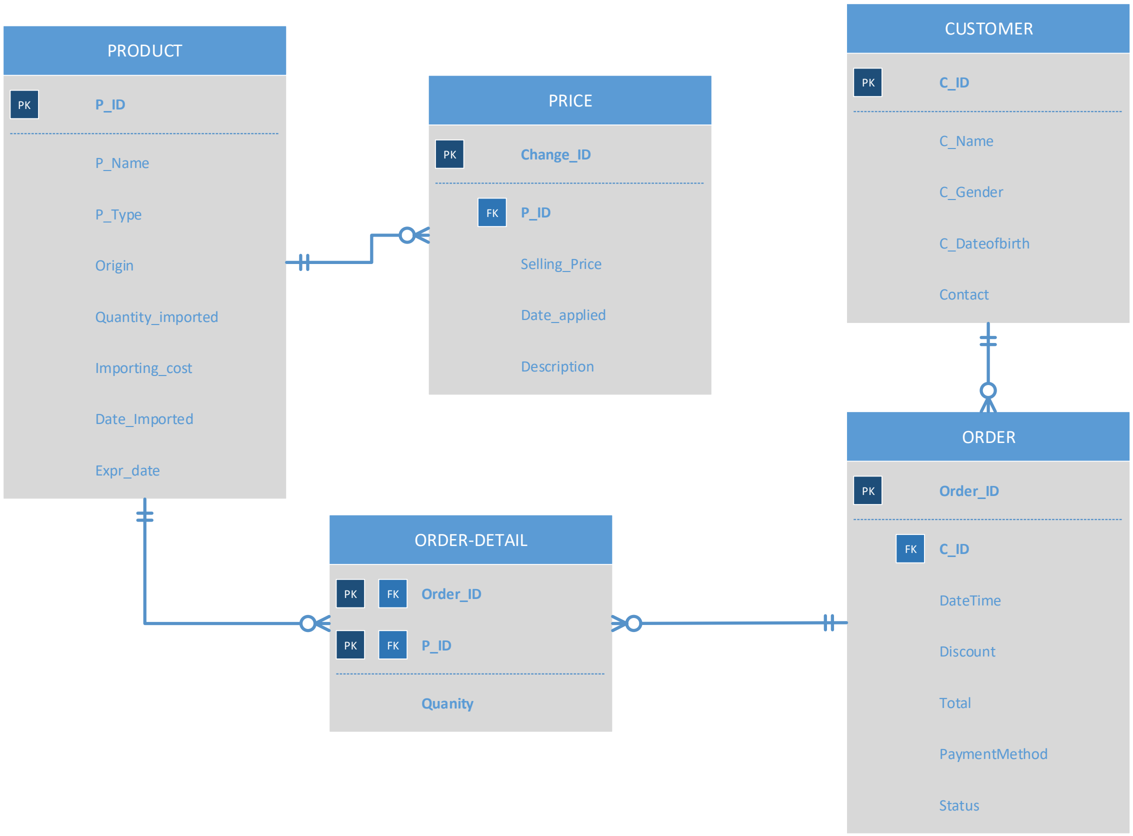
- STAFF (S_ID, S_Name, S_Gender, S_Dateofbirth, Address, Contact, Hire_date, Job)
- WORK (Record_ID, S_ID, Start_Time, End_Time)
  - Fk: S_ID ➔ STAFF
- SALARY (Trans_ID, S_ID, Trans_Date, Amount, Method)
  - Fk: S_ID ➔ STAFF
- PRODUCT (P_ID, P_Name, P_Type, Origin, Quantity_imported, Importing_cost, Date_Imported, Expr_Date)
- PRICE(Price_ID, P_ID, NewSellingPrice, Date_applied, Description)
  - Fk: P_ID ➔ PRODUCT
- CUSTOMER(C_ID, C_Name, C_Gender, C_Dateofbirth, Contact)
- ORDERS(Order_ID, C_ID, DateTime, Discount, Total, PaymentMethod, Status)
  - Fk: C_ID ➔ CUSTOMER
- ORDER-DETAIL(Order_ID, P_ID, Quantity)
  - Fk: Order_ID ➔ ORDER
    P_ID ➔ PRODUCT



SALARY
- PK Trans_ID
- FK S_ID
- Trans_Date
- Amount
- Method

STAFF
- PK S_ID
- S_Name
- S_Gender
- S_Dateofbirth
- Address
- Contact
- Hire_date
- Job

WORK
- PK Record_ID
- FK S_ID
- Start_Time
- End_Time



PRODUCT
- PK P_ID
- P_Name
- P_Type
- Origin
- Quantity_imported
- Importing_cost
- Date_Imported
- Expr_date

PRICE
- PK Change_ID
- FK P_ID
- Selling_Price
- Date_applied
- Description

CUSTOMER
- PK C_ID
- C_Name
- C_Gender
- C_Dateofbirth
- Contact

ORDER
- PK Order_ID
- FK C_ID
- DateTime
- Discount
- Total
- PaymentMethod
- Status

ORDER-DETAIL
- PK FK Order_ID
- PK FK P_ID
- Quanity

Part III. ORACLE IMPLEMENTATION

- Sql scripts:

```
DROP TABLE ORDER_DETAIL CASCADE CONSTRAINTS;
DROP TABLE ORDERS CASCADE CONSTRAINTS;
DROP TABLE PRICE CASCADE CONSTRAINTS;
DROP TABLE CUSTOMER CASCADE CONSTRAINTS;
DROP TABLE WORK CASCADE CONSTRAINTS;
DROP TABLE SALARY CASCADE CONSTRAINTS;
DROP TABLE STAFF CASCADE CONSTRAINTS;
DROP TABLE PRODUCT CASCADE CONSTRAINTS;

CREATE TABLE STAFF(
  S_ID NUMBER(3) ,
  S_Name VARCHAR(30) NOT NULL,
  S_Gender VARCHAR(1),
  S_Dateofbirth DATE,
  Address VARCHAR(50),
  Contact VARCHAR(40) NOT NULL,
  Hire_date DATE ,
  Job VARCHAR (10) not null,
  constraint STAFF_PK primary key (S_ID),
  constraint S_Gender_Validation CHECK(S_GENDER IN ('F','M','L','G','B','T','Q'))
);


CREATE TABLE WORK (
  Record_ID number,
  S_ID number not null,
  Start_Time TIMESTAMP not null,
  End_Time TIMESTAMP not null,
  constraint work_pk primary key (Record_ID),
  constraint work_SID_fk foreign key (S_ID) REFERENCES STAFF(S_ID),
  CONSTRAINT Work_time_validation CHECK (
      to_char(Start_Time, 'YYYY-MM-DD HH24:MI:SS')<to_char(End_Time, 'YYYY-
MM-DD HH24:MI:SS')
        )
);
```

```
CREATE TABLE SALARY (
  Trans_ID NUMBER,
  S_ID NUMBER,
  Trans_Date TIMESTAMP NOT NULL,
  Amount NUMBER NOT NULL,
  Method VARCHAR(20) NOT NULL,
  CONSTRAINT SALARY_PK PRIMARY KEY(Trans_ID),
  CONSTRAINT SALARY_S_ID_fK foreign key (S_ID) REFERENCES STAFF(S_ID),
  CONSTRAINT SALSARY_AMOUNT_VALIDATION CHECK (Amount>=0)
);


CREATE TABLE PRODUCT(
      P_ID number,
      P_Name varchar(50) NOT NULL,
      P_Type VARCHAR(12),
      Origin VARCHAR(12) NOT NULL,
      Quantity_imported NUMBER NOT NULL,
      Importing_cost NUMBER NOT NULL,
      Date_Imported DATE NOT NULL,
      Expr_Date DATE,
      CONSTRAINT Product_PK primary key(P_ID),
      constraint Valid_long_life_of_product CHECK (Date_Imported < Expr_date)
);

CREATE TABLE PRICE(
    Price_ID NUMBER,
    P_ID NUMBER NOT NULL,
    NewSellingPrice NUMBER NOT NULL,
    Date_applied DATE NOT NULL,
    Description VARCHAR(30),
    CONSTRAINT PRICE_PK PRIMARY KEY (Price_ID),
    CONSTRAINT PRICE_PID_FK FOREIGN KEY (P_ID) REFERENCES
PRODUCT(P_ID)
);
CREATE TABLE CUSTOMER(
    C_ID NUMBER,
    C_Name VARCHAR(30),
    C_Gender varchar(1),
    C_Dateofbirth DATE,
    Contact VARCHAR(40),
    CONSTRAINT CUSTOMER_PK PRIMARY KEY (C_ID),
    constraint c_Gender_Validation CHECK(C_GENDER IN ( null,
'F','M','L','G','B','T','Q'))
);
```

```
CREATE TABLE ORDERS(
   Order_ID NUMBER,
   C_ID NUMBER,
   DateTime TIMESTAMP not null,
   Discount NUMBER default 0,
   Total NUMBER,
   PaymentMethod VARCHAR(12) NOT NULL,
   Status varchar(10),
   CONSTRAINT ORDER_PK PRIMARY KEY(Order_ID),
   CONSTRAINT ORDER_CID_FK foreign key (C_ID) REFERENCES
CUSTOMER(C_ID),
   CONSTRAINT ORDER_VALIDATION CHECK (
      (PaymentMethod IN ('Cash','Check','Paypal','Bitcoin','VISA','Master Card','Other'))
      AND (DISCOUNT<=TOTAL)
      )
);
CREATE TABLE ORDER_DETAIL(
   Order_ID NUMBER,
   P_ID NUMBER,
   Quantity NUMBER NOT NULL,
   CONSTRAINT ORDERDETAIL_PK PRIMARY KEY(Order_ID, P_ID),
   CONSTRAINT ORDERDETAIL_ODERID_FK FOREIGN KEY(Order_ID)
REFERENCES ORDERS(Order_ID),
   CONSTRAINT ORDERDETAIL_PID_FK FOREIGN KEY(P_ID) REFERENCES
PRODUCT(P_ID),
   CONSTRAINT Quantity_VALIDATION CHECK(Quantity>0)
);
COMMIT;

INSERT INTO STAFF VALUES
(00,'Quan Tran','M','19-Jul-1992','GA','1-460-997-4423','09-Oct-2015','IT-System');
INSERT INTO STAFF VALUES
(01,'Lillith Velazquez','F','23-Jan-1990','GA','1-698-108-8412','08-Sep-2011','SALES');
INSERT INTO STAFF VALUES
(02,'Jeanette Solis','F','06-Apr-1990','GA','1-698-418-2676','22-Aug-2011','SALES');
INSERT INTO STAFF VALUES
(03,'Yeo Holder','F','26-Nov-1993','GA','1-616-240-1791','18-Oct-2011','SALES');
INSERT INTO STAFF VALUES
(04,'Leilani Evans','F','09-Apr-1991','GA','1-862-510-5434','10-Dec-2010','SALES');
INSERT INTO STAFF VALUES
(05,'Dacey Potts','F','04-Aug-1993','GA','1-669-879-2506','03-Nov-2010','SALES');
INSERT INTO STAFF VALUES
(06,'Brent Farrell','M','12-Jul-1993','GA','1-802-968-3763','04-Aug-2011','SALES');
INSERT INTO STAFF VALUES
(07,'Ciaran Workman','M','20-Mar-1991','GA','1-468-465-0727','03-Jan-2011','SALES');
```

```
INSERT INTO STAFF VALUES
(08,'Marwin Jeff','M','19-Jul-1992','GA','1-460-997-4423','09-Oct-2015','SALES');
INSERT INTO STAFF VALUES
(09,'Yoshio Ware','M','27-Oct-1994','GA','1-658-250-2256','02-Jun-2010','MANAGER');
INSERT INTO STAFF VALUES
(10,'Gary Little','M','30-May-1994','GA','1-655-118-3548','05-Jul-2009','MANAGER');


INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(1,1,TO_TIMESTAMP('11/12/2015 07:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/12/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'));
INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(2,2,TO_TIMESTAMP('11/12/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/12/2015 22:00:00', 'mm/dd/YYYY
HH24:MI:SS'));
INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(3,3,TO_TIMESTAMP('11/13/2015 07:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/13/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'));
INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(4,4,TO_TIMESTAMP('11/13/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/13/2015 22:00:00', 'mm/dd/YYYY
HH24:MI:SS'));
INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(5,5,TO_TIMESTAMP('11/14/2015 07:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/14/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'));
INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(6,6,TO_TIMESTAMP('11/14/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/14/2015 22:00:00', 'mm/dd/YYYY
HH24:MI:SS'));
INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(7,7,TO_TIMESTAMP('11/15/2015 07:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/15/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'));
INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(8,8,TO_TIMESTAMP('11/15/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/15/2015 22:00:00', 'mm/dd/YYYY
HH24:MI:SS'));
INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(9,9,TO_TIMESTAMP('11/12/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/12/2015 22:00:00', 'mm/dd/YYYY
HH24:MI:SS'));
```

```
INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(10,10,TO_TIMESTAMP('11/14/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/14/2015 22:00:00', 'mm/dd/YYYY
HH24:MI:SS'));

INSERT INTO WORK(
  SELECT (SELECT MAX(Record_ID) from work)+Record_ID,S_ID,

(TO_TIMESTAMP(TO_CHAR(TO_DATE(to_char(START_TIME,'mm/dd/YYYY'),'m
m/dd/YYYY')+4, 'mm/dd/YYYY')||' '||
            TO_CHAR(START_TIME,'HH24:MI:SS'),'mm/dd/YYYY HH24:MI:SS')),

(TO_TIMESTAMP(TO_CHAR(TO_DATE(to_char(END_TIME,'mm/dd/YYYY'),'mm/
dd/YYYY')+4, 'mm/dd/YYYY')||' '||
            TO_CHAR((End_time),'HH24:MI:SS'),'mm/dd/YYYY HH24:MI:SS'))
  from work
  where RECORD_ID>0 and S_ID>0);

INSERT INTO WORK(
  SELECT (SELECT MAX(Record_ID) from work)+Record_ID,S_ID,
(TO_TIMESTAMP(TO_CHAR(TO_DATE(to_char(START_TIME,'mm/dd/YYYY'),'m
m/dd/YYYY')+8, 'mm/dd/YYYY')||' '||
            TO_CHAR(START_TIME,'HH24:MI:SS'),'mm/dd/YYYY HH24:MI:SS')),
(TO_TIMESTAMP(TO_CHAR(TO_DATE(to_char(END_TIME,'mm/dd/YYYY'),'mm/
dd/YYYY')+8, 'mm/dd/YYYY')||' '||
            TO_CHAR((End_time),'HH24:MI:SS'),'mm/dd/YYYY HH24:MI:SS'))
  from work
  where RECORD_ID>0 and S_ID>0);

INSERT INTO WORK (Record_ID,S_ID,Start_Time,End_Time) VALUES
(0,00,TO_TIMESTAMP('11/10/2015 07:00:00', 'mm/dd/YYYY
HH24:MI:SS'),TO_TIMESTAMP('11/12/2015 14:00:00', 'mm/dd/YYYY
HH24:MI:SS'));

INSERT INTO SALARY
(SELECT RECORD_ID,
      S_ID,
      END_TIME,
      (DECODE(JOB,'SALES',10,'MANAGER',08,'IT-System',100,0)
       * (EXTRACT (DAY    FROM (End_Time-Start_Time))*24
        + EXTRACT (HOUR   FROM (End_Time-Start_Time))
        + EXTRACT (MINUTE FROM (End_Time-Start_Time))/60)),
      'Cash'
      from WORK JOIN STAFF USING (S_ID)
      where RECORD_ID not in (SELECT TRANS_ID FROM SALARY));
```

```
commit;

INSERT INTO PRODUCT VALUES
(1,'Sweetart','Candy','US',50,150.00,'01-Oct-2015','01-Oct-2017');
INSERT INTO PRODUCT VALUES
(2,'M and M','Candy','US',50,150.00,'01-Oct-2015','01-Oct-2017');
INSERT INTO PRODUCT VALUES
(3,'Chocolate Oreo','Cookie','US',50,150.00,'01-Oct-2015','01-Oct-2017');
INSERT INTO PRODUCT VALUES
(4,'Ocreamo','Cookie','US',50,150.00,'01-Oct-2015','01-Oct-2017');
INSERT INTO PRODUCT VALUES
(5,'Travelling charger for Apple Iphone','Accessories','US',20,400.00,'01-Oct-2015',null);
INSERT INTO PRODUCT VALUES
(6,'Travelling charger for Samsung mobile','Accessories','US',20,320.00,'01-Oct-2015',null);
INSERT INTO PRODUCT VALUES
(7,'Travelling charger for Sony mobile','Accessories','US',10,150.00,'01-Oct-2015',null);
INSERT INTO PRODUCT VALUES
(8,'Sandisk Extreme pro 3 32gb SDcard','Accessories','US',15,600.00,'01-Oct-2015',null);
INSERT INTO PRODUCT VALUES
(9,'Sandisk Pixtor 64gb microSDcard','Accessories','US',15,1500.00,'01-Oct-2015',null);
INSERT INTO PRODUCT VALUES
(10,'Sandisk 128gb SDcard','Accessories','US',5,400.00,'01-Oct-2015',null);

INSERT INTO PRICE(
SELECT P_ID, P_ID,
(IMPORTING_COST*1.20/QUANTITY_IMPORTED),DATE_IMPORTED,'Initial Sale
price' FROM PRODUCT);


INSERT INTO CUSTOMER VALUES
(1,'Ivor G. Alston','M','08-Jan-1989','1-960-973-5181');
INSERT INTO CUSTOMER VALUES
(2,'Lacota R. Sexton','M','06-Mar-1989','1-478-254-9765');
INSERT INTO CUSTOMER VALUES
(3,'Tashya T. Welch','M','19-May-1996','1-806-982-3024');
INSERT INTO CUSTOMER VALUES
(4,'Quemby Z. Allen','F','08-Apr-1990','1-973-100-1837');
INSERT INTO CUSTOMER VALUES
(5,'Beatrice J. Bright','F','07-Mar-1989','1-477-670-9377');
INSERT INTO CUSTOMER VALUES
(6,'Hermione G. Burt','M','13-Oct-1989','1-658-760-2376');
INSERT INTO CUSTOMER VALUES
(7,'Zenia J. Cole','F','13-Oct-1990','1-588-378-3985');
INSERT INTO CUSTOMER VALUES
(8,'Eric P. Hall','M','03-Apr-1996','1-274-860-4563');
```

```
INSERT INTO CUSTOMER VALUES
(9,'Uta R. Hahn','M','03-Jun-1992','1-499-637-5703');
INSERT INTO CUSTOMER VALUES
(10,'Tate T. Cabrera','M','10-Jan-1985','1-791-193-0806');

INSERT INTO ORDERS VALUES (1,1,TO_TIMESTAMP('17-Nov-2015 13.01.05',
'dd-Mon-YYYY HH24:MI:SS'),0,0,'VISA','Closed');
INSERT INTO ORDERS VALUES (2,2,TO_TIMESTAMP('14-Nov-2015 18.43.09',
'dd-Mon-YYYY HH24:MI:SS'),0,0,'VISA','Closed');
INSERT INTO ORDERS VALUES (3,3,TO_TIMESTAMP('16-Nov-2015 19.24.28',
'dd-Mon-YYYY HH24:MI:SS'),0,0,'VISA','Closed');
INSERT INTO ORDERS VALUES (4,4,TO_TIMESTAMP('18-Nov-2015 12.19.57',
'dd-Mon-YYYY HH24:MI:SS'),0,0,'VISA','Closed');
INSERT INTO ORDERS VALUES (5,5,TO_TIMESTAMP('15-Nov-2015 20.16.42',
'dd-Mon-YYYY HH24:MI:SS'),0,0,'VISA','Closed');
INSERT INTO ORDERS VALUES (6,6,TO_TIMESTAMP('17-Nov-2015 08.37.57',
'dd-Mon-YYYY HH24:MI:SS'),0,0,'VISA','Closed');
INSERT INTO ORDERS VALUES (7,7,TO_TIMESTAMP('13-Nov-2015 07.56.38',
'dd-Mon-YYYY HH24:MI:SS'),0,0,'VISA','Closed');
INSERT INTO ORDERS VALUES (8,8,TO_TIMESTAMP('19-Nov-2015 11.56.43',
'dd-Mon-YYYY HH24:MI:SS'),0,0,'VISA','Closed');
INSERT INTO ORDERS VALUES (9,9,TO_TIMESTAMP('20-Nov-2015 17.56.43',
'dd-Mon-YYYY HH24:MI:SS'),0,0,'VISA','Closed');
INSERT INTO ORDERS VALUES (10,10,TO_TIMESTAMP('22-Nov-2015 13.56.43',
'dd-Mon-YYYY HH24:MI:SS'),0,0,'VISA','Closed');

INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (10,5,1);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (5,1,2);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (2,5,2);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (4,5,1);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (2,10,1);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (7,1,1);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (3,3,1);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (2,4,1);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (1,2,1);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (3,2,2);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (4,8,1);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (6,5,2);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (7,9,2);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (6,10,1);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (9,1,1);
INSERT INTO ORDER_DETAIL (Order_ID,P_ID,Quantity) VALUES (1,4,2);
```

```
UPDATE ORDERS ord SET TOTAL = (
            SELECT sum(o_d.QUANTITY*prc.NEWSELLINGPRICE)
            from ORDER_DETAIL o_d JOIN PRICE prc USING (P_ID)
            WHERE (o_d.ORDER_ID=ord.ORDER_ID)
            and (prc.DATE_APPLIED >= ALL (SELECT DATE_APPLIED FROM PRICE
WHERE DATE_APPLIED<=ord.DATETIME))
        );
commit;
```

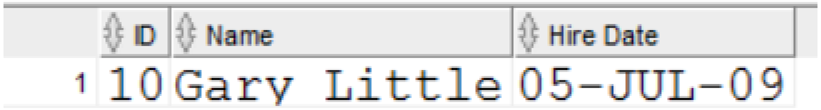
- Feedback:

```
Table ORDER_DETAIL dropped.
Table ORDERS dropped.
Table PRICE dropped.
Table CUSTOMER dropped.
Table WORK dropped.
Table SALARY dropped.
Table STAFF dropped.
Table PRODUCT dropped.
Table STAFF created.
Table WORK created.
Table SALARY created.
Table PRODUCT created.
Table PRICE created.
Table CUSTOMER created.
Table ORDERS created.
Table ORDER_DETAIL created.
Commit complete.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
10 rows inserted.
20 rows inserted.
1 row inserted.
41 rows inserted.
Commit complete.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
10 rows inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
10 rows updated.

Commit complete.
```

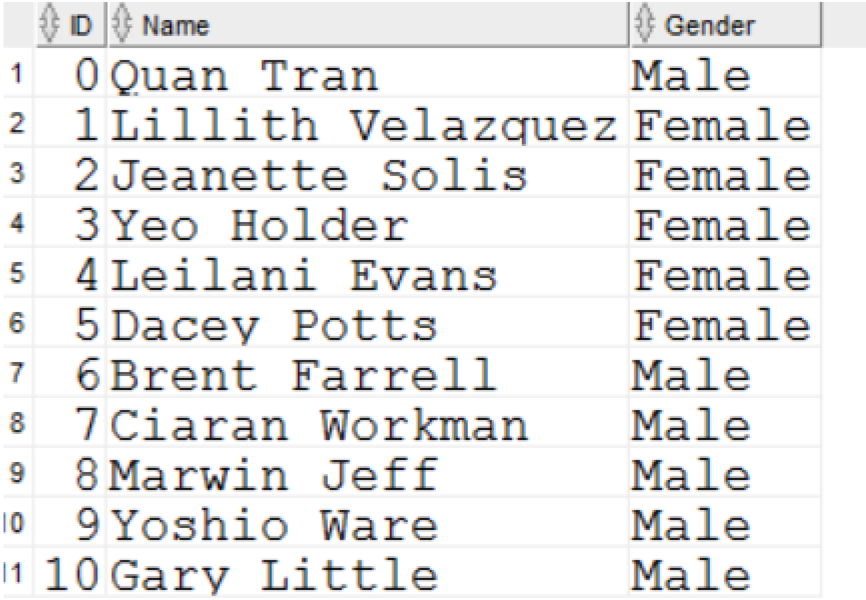
Part IV: QUERIES

```
/* Query 1 - Compound condition
 * LISTS ALL THE MANAGERS, WHO HAVE BEEN WORKING AT THE STORE BEFORE 2010.
 * Show the id, name and hire_date.
 */
SELECT
   S_ID as "ID",
   S_NAME as "Name",
   HIRE_DATE as "Hire Date"
FROM STAFF
WHERE (EXTRACT(YEAR FROM HIRE_DATE)<2010)
     AND (JOB='MANAGER');
```



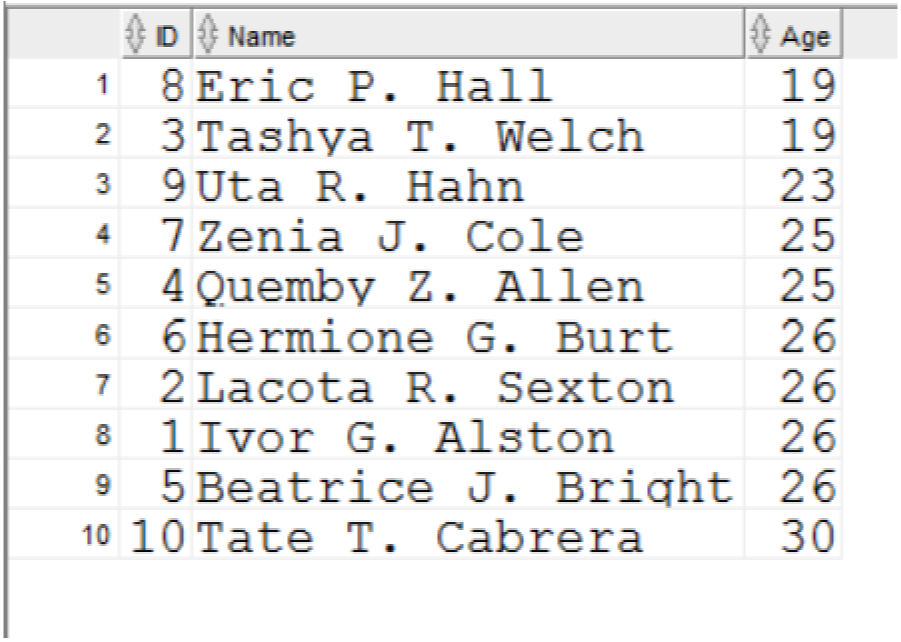
	ID	Name	Hire Date
1	10	Gary Little	05-JUL-09

```
/* Query 2 - Case or decode statement
 * List all the staffs working at the store,
 *  show 'Male' and 'Female' for gender, instead of just original F or M.
 * Show the ID, Name and Gender.
 */
SELECT
   S_ID as "ID",
   S_NAME as "Name",
   DECODE(S_GENDER,'M','Male','F','Female','Other') as "Gender"
FROM STAFF;
```



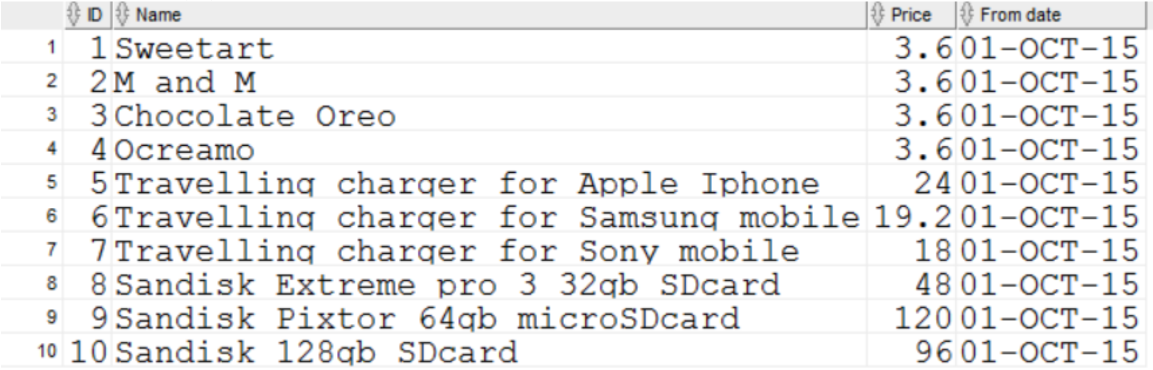
	ID	Name	Gender
1	0	Quan Tran	Male
2	1	Lillith Velazquez	Female
3	2	Jeanette Solis	Female
4	3	Yeo Holder	Female
5	4	Leilani Evans	Female
6	5	Dacey Potts	Female
7	6	Brent Farrell	Male
8	7	Ciaran Workman	Male
9	8	Marwin Jeff	Male
10	9	Yoshio Ware	Male
11	10	Gary Little	Male

```
/* Query 3 - Build-in function
 * Show the ID, Name and Age for all the customers visited the store. Order by Age
 */
SELECT
   C_ID as "ID",
   C_NAME as "Name",
   Trunc(Months_between(SYSDATE,C_DATEOFBIRTH)/12) as "Age"
FROM CUSTOMER
ORDER BY 3;
```



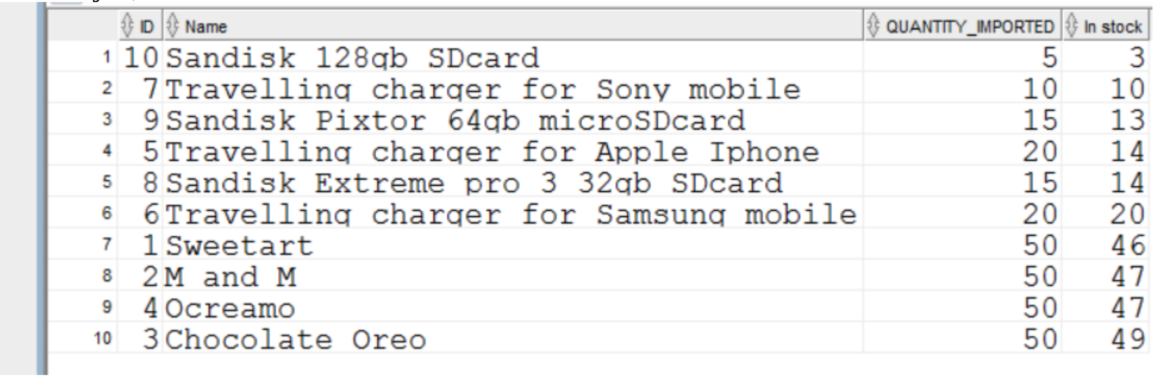
	ID	Name	Age
1	8	Eric P. Hall	19
2	3	Tashya T. Welch	19
3	9	Uta R. Hahn	23
4	7	Zenia J. Cole	25
5	4	Quemby Z. Allen	25
6	6	Hermione G. Burt	26
7	2	Lacota R. Sexton	26
8	1	Ivor G. Alston	26
9	5	Beatrice J. Bright	26
10	10	Tate T. Cabrera	30

```
/* Query 4: JOIN
 * For every product, show the id, Name and along with its price history.
 */
SELECT
    prd.P_ID "ID",
    prd.P_NAME "Name",
    prc.NEWSELLINGPRICE "Price",
    prc.DATE_APPLIED "From date"
FROM PRODUCT prd JOIN PRICE prc ON (prd.P_ID=prc.P_ID)
ORDER BY 1,4;
```



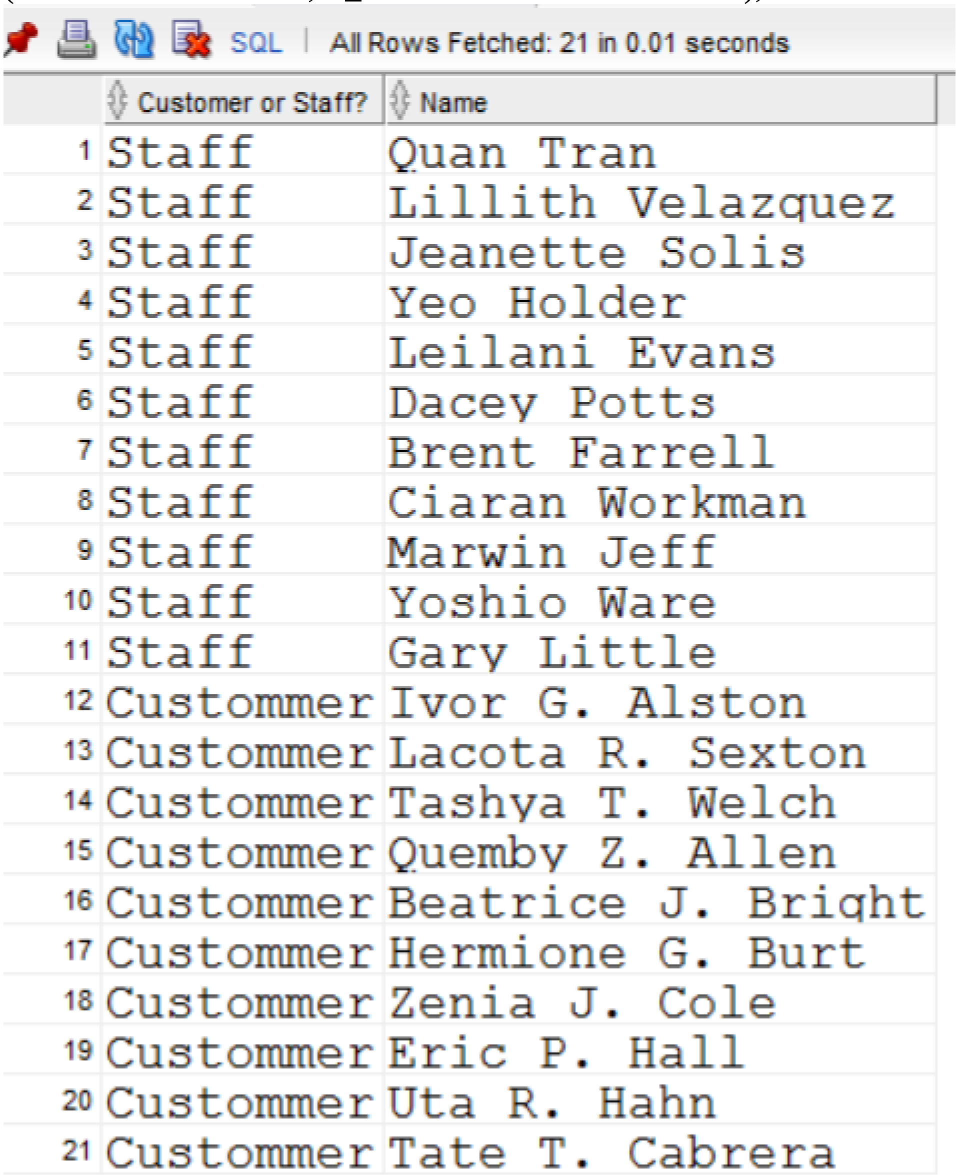
	ID	Name	Price	From date
1	1	Sweetart	3.6	01-OCT-15
2	2	M and M	3.6	01-OCT-15
3	3	Chocolate Oreo	3.6	01-OCT-15
4	4	Ocreamo	3.6	01-OCT-15
5	5	Travelling charger for Apple Iphone	24	01-OCT-15
6	6	Travelling charger for Samsung mobile	19.2	01-OCT-15
7	7	Travelling charger for Sony mobile	18	01-OCT-15
8	8	Sandisk Extreme pro 3 32gb SDcard	48	01-OCT-15
9	9	Sandisk Pixtor 64gb microSDcard	120	01-OCT-15
10	10	Sandisk 128gb SDcard	96	01-OCT-15

```
/* Query 5: SINGLE VALUE SUBQUERY
 * For every product, show the id, Name, the quantity imported
 * and how many currently left. Order by the number in stock
 */
SELECT
   prd.P_ID "ID",
   prd.P_NAME "Name",
   prd.QUANTITY_IMPORTED,
   prd.QUANTITY_IMPORTED-NVL((SELECT SUM(od.QUANTITY) FROM
ORDER_DETAIL od WHERE od.P_ID=prd.P_ID),0)
      AS "In stock"
FROM PRODUCT prd
order by 4;
```



	ID	Name	QUANTITY_IMPORTED	In stock
1	10	Sandisk 128gb SDcard	5	3
2	7	Travelling charger for Sony mobile	10	10
3	9	Sandisk Pixtor 64gb microSDcard	15	13
4	5	Travelling charger for Apple Iphone	20	14
5	8	Sandisk Extreme pro 3 32gb SDcard	15	14
6	6	Travelling charger for Samsung mobile	20	20
7	1	Sweetart	50	46
8	2	M and M	50	47
9	4	Ocreamo	50	47
10	3	Chocolate Oreo	50	49

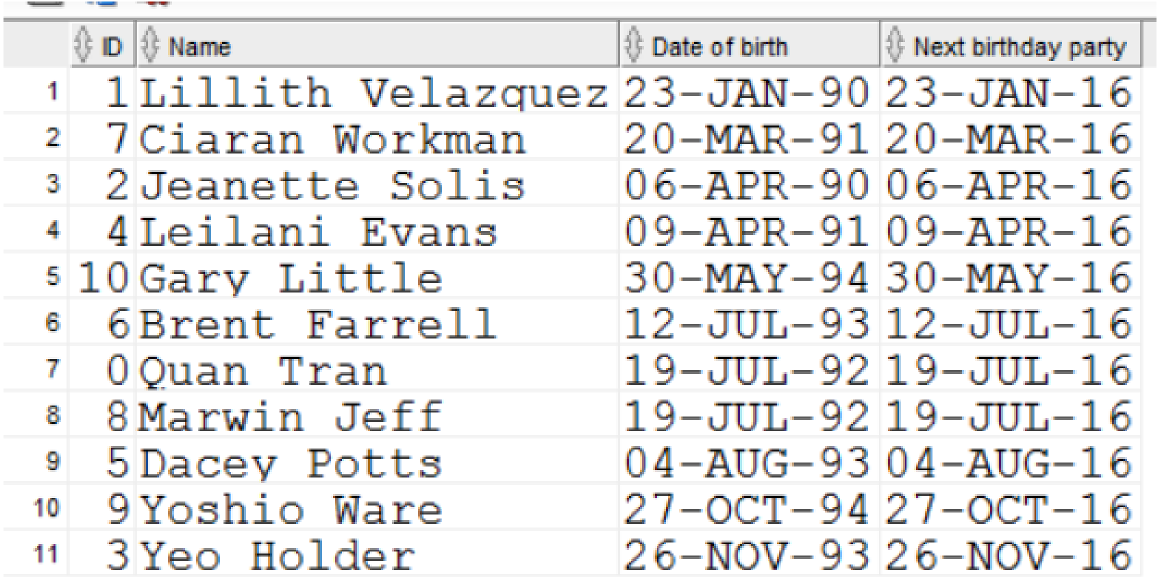
```
/* Query 6: Set operator
 * Show the name of all the staffs and customers, also determine who it is.
 * Staffs are shown first.
 */
(SELECT 'Staff' as "Customer or Staff?", S_NAME as "Name" FROM STAFF)
UNION ALL
(SELECT 'Custommer' , C_NAME  FROM CUSTOMER);
```



All Rows Fetched: 21 in 0.01 seconds

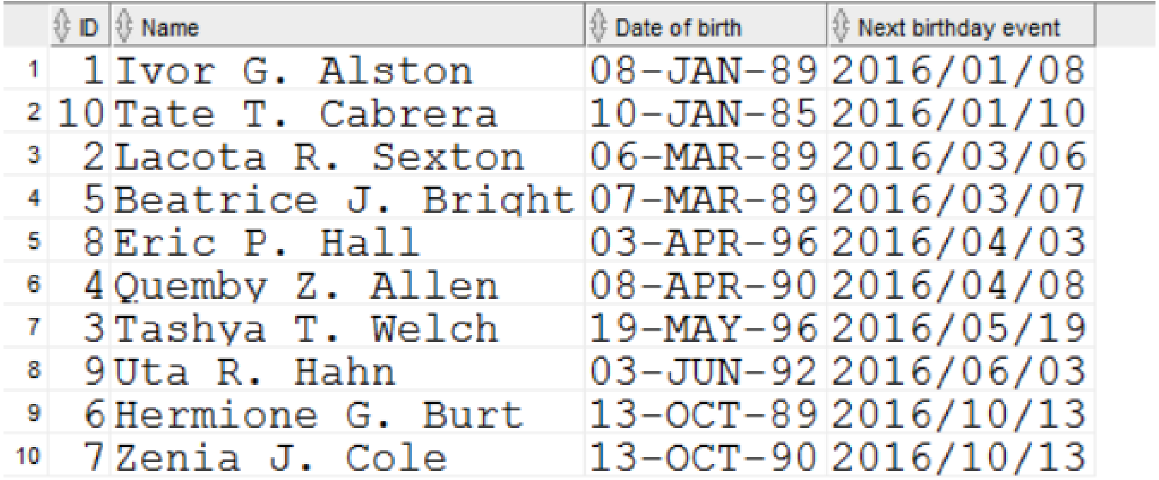
	Customer or Staff?	Name
1	Staff	Quan Tran
2	Staff	Lillith Velazquez
3	Staff	Jeanette Solis
4	Staff	Yeo Holder
5	Staff	Leilani Evans
6	Staff	Dacey Potts
7	Staff	Brent Farrell
8	Staff	Ciaran Workman
9	Staff	Marwin Jeff
10	Staff	Yoshio Ware
11	Staff	Gary Little
12	Custommer	Ivor G. Alston
13	Custommer	Lacota R. Sexton
14	Custommer	Tashya T. Welch
15	Custommer	Quemby Z. Allen
16	Custommer	Beatrice J. Bright
17	Custommer	Hermione G. Burt
18	Custommer	Zenia J. Cole
19	Custommer	Eric P. Hall
20	Custommer	Uta R. Hahn
21	Custommer	Tate T. Cabrera

```
/* Query 7: Date function
 * Show the date of next birthday party for every staff.
 * Order by the next party.
 */
 SELECT
   S_ID as "ID",
   S_NAME as "Name",
   S_DATEOFBIRTH "Date of birth",
   Add_months(S_DATEOFBIRTH,
           (12*(trunc(months_between(Sysdate-1, S_DATEOFBIRTH)/12)+1))
           )   as "Next birthday party"
FROM STAFF
ORDER BY 4 ;
```



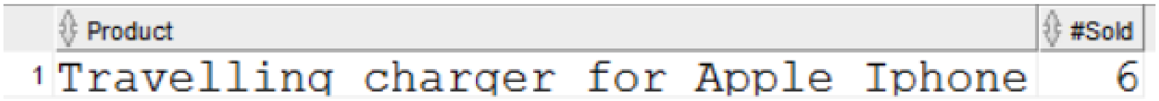
	ID	Name	Date of birth	Next birthday party
1	1	Lillith Velazquez	23-JAN-90	23-JAN-16
2	7	Ciaran Workman	20-MAR-91	20-MAR-16
3	2	Jeanette Solis	06-APR-90	06-APR-16
4	4	Leilani Evans	09-APR-91	09-APR-16
5	10	Gary Little	30-MAY-94	30-MAY-16
6	6	Brent Farrell	12-JUL-93	12-JUL-16
7	0	Quan Tran	19-JUL-92	19-JUL-16
8	8	Marwin Jeff	19-JUL-92	19-JUL-16
9	5	Dacey Potts	04-AUG-93	04-AUG-16
10	9	Yoshio Ware	27-OCT-94	27-OCT-16
11	3	Yeo Holder	26-NOV-93	26-NOV-16

```
/* Query 8 - DATA CONVERSION function
* Show the date of next birthday event for every customer in format YYYY/MM/DD
* Order by the next birthday.
*/
 SELECT
    C_ID as "ID",
    C_NAME as "Name",
    C_DATEOFBIRTH "Date of birth",
    To_char(Add_months(C_DATEOFBIRTH,
              (12*(trunc(months_between(Sysdate-1, C_DATEOFBIRTH)/12)+1))
              ),'YYYY/MM/DD')   as "Next birthday event"
FROM CUSTOMER
ORDER BY 4 ;
```



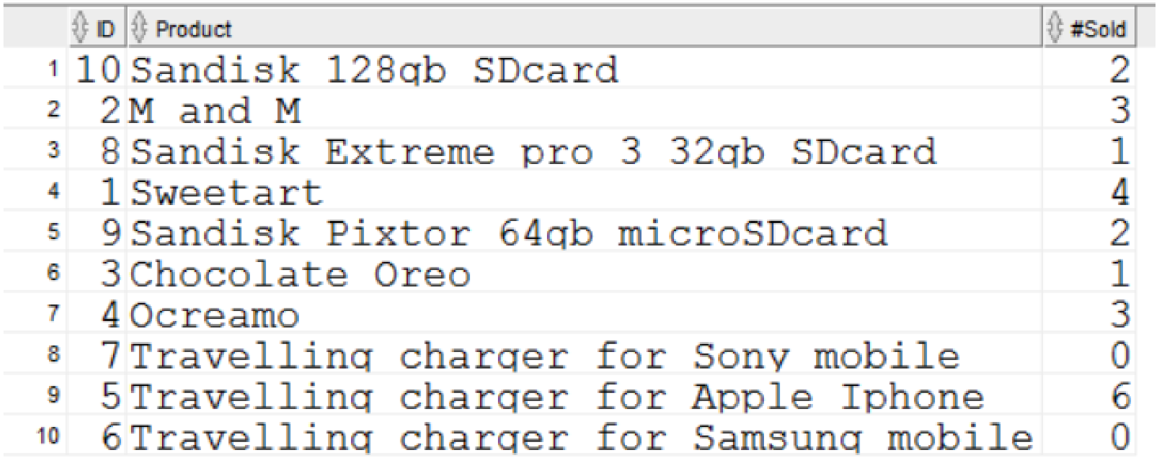
	ID	Name	Date of birth	Next birthday event
1	1	Ivor G. Alston	08-JAN-89	2016/01/08
2	10	Tate T. Cabrera	10-JAN-85	2016/01/10
3	2	Lacota R. Sexton	06-MAR-89	2016/03/06
4	5	Beatrice J. Bright	07-MAR-89	2016/03/07
5	8	Eric P. Hall	03-APR-96	2016/04/03
6	4	Quemby Z. Allen	08-APR-90	2016/04/08
7	3	Tashya T. Welch	19-MAY-96	2016/05/19
8	9	Uta R. Hahn	03-JUN-92	2016/06/03
9	6	Hermione G. Burt	13-OCT-89	2016/10/13
10	7	Zenia J. Cole	13-OCT-90	2016/10/13

```
/* Query 9 - Multiple value subquery
 * Show the best selling product in store.
 */
SELECT P_NAME "Product", sum(ORDER_DETAIL.QUANTITY) "#Sold"
FROM PRODUCT join ORDER_DETAIL ON (PRODUCT.P_ID =
ORDER_DETAIL.P_ID)
GROUP BY PRODUCT.P_ID,PRODUCT.P_NAME
HAVING SUM(QUANTITY)>=ALL ( SELECT SUM(QUANTITY) FROM
ORDER_DETAIL GROUP BY P_ID);
```



	Product	#Sold
1	Travelling charger for Apple Iphone	6

```
/* Query 10 - Outer join
 * Show Product and the number of units sold.
 */
SELECT
   prd.P_ID "ID",
   prd.P_NAME "Product",
   NVL(SUM(od.QUANTITY),0) "#Sold"
FROM PRODUCT prd LEFT OUTER JOIN ORDER_DETAIL od ON (prd.P_ID =
od.P_ID)
GROUP BY
prd.P_ID,prd.P_NAME;
```



	ID	Product	#Sold
1	10	Sandisk 128gb SDcard	2
2	2	M and M	3
3	8	Sandisk Extreme pro 3 32gb SDcard	1
4	1	Sweetart	4
5	9	Sandisk Pixtor 64gb microSDcard	2
6	3	Chocolate Oreo	1
7	4	Ocreamo	3
8	7	Travelling charger for Sony mobile	0
9	5	Travelling charger for Apple Iphone	6
10	6	Travelling charger for Samsung mobile	0

Part V: SCRIPTS

- User will enter an id of a staff, then show the staff profile, along with all his/her works, salary.

```
SQL> get C:\Oracle\Project\script-part5.sql
  1  SET PAGESIZE 20
  2  SET LINESIZE 100
  3  SET FEEDBACK OFF
  4  SET PAUSE OFF
  5  TTITLE ‘STAFF Report’
  6  BTITLE ‘END PROFILE’
  7  -- SQL statement
  8  SELECT
  9       S_ID "ID",
 10       S_Name "Name",
 11       JOB "Job",
 12       Trunc(months_between(Sysdate, S_DATEOFBIRTH)/12) "Age",
 13       Address,
 14       Contact
 15  FROM STAFF
 16  WHERE STAFF.S_ID='&&ENTER_ID';
 17  TTITLE 'WORK TRACKING'
 18  BTITLE ‘END WORKS’
 19  SELECT
 20      RECORD_ID "Record No",
 21      To_char(START_TIME,'mm/dd/yyyy hh24:mi:ss') "Start time",
 22      To_char(END_TIME,'mm/dd/yyyy hh24:mi:ss') "End time"
 23  FROM WORK
 24  WHERE S_ID='&ENTER_ID';
 25  COL Amount FORMAT $99,999.99
 26  TTITLE ‘SALARY’
 27  BTITLE ‘END SALARY’
 28  SELECT
 29       Trans_ID "Transaction No",
 30       To_char(Trans_Date,'mm/dd/yyyy') "Date",
 31       Amount,
 32       Method
 33  FROM SALARY
 34* WHERE S_ID='&ENTER_ID';
 35  @ C:\Oracle\Project\script-part5.sql
Enter value for enter_id: 3
old   9: WHERE STAFF.S_ID='&&ENTER_ID'
new   9: WHERE STAFF.S_ID='3'
```

```
‘STAFFReport’
      ID Name                          Job              Age
---------- ------------------------------ ---------- ----------
ADDRESS
--------------------------------------------------
CONTACT
----------------------------------------
         3 Yeo Holder                    SALES              22
GA
1-616-240-1791



‘ENDPROFILE’
old   6: WHERE S_ID='&ENTER_ID'
new   6: WHERE S_ID='3'

Tue Dec 08                                          page    1
                              WORK TRACKING

 Record No Start time          End time
---------- ------------------- -------------------
         3 11/13/2015 07:00:00 11/13/2015 14:00:00
        13 11/17/2015 07:00:00 11/17/2015 14:00:00
        23 11/21/2015 07:00:00 11/21/2015 14:00:00
        33 11/25/2015 07:00:00 11/25/2015 14:00:00




‘ENDWORKS’
old   7: WHERE S_ID='&ENTER_ID'
new   7: WHERE S_ID='3'

Tue Dec 08                                          page    1
                              ‘SALARY’

Transaction No Date            AMOUNT METHOD
-------------- ---------- ----------- --------------------
             3 11/13/2015      $70.00 Cash
            13 11/17/2015      $70.00 Cash
            23 11/21/2015      $70.00 Cash
            33 11/25/2015      $70.00 Cash




‘ENDSALARY’
SQL> spool off
```