ASSIGNMENT - 3

Procedural SQL

CS3003 DATABASE MANAGEMENT SYSTEM

Submitted by: Vidushi Yadav, AU20B1013

PROBLEM STATEMENT

For this assignment, students must use the same database schema which they used in assignment 2. Follow the below steps to complete the assignment.

- Identify the tables on which stored procedures can be useful and write at least 3 stored procedures on it. Justify your table selected and query at the time of evaluation.
- Write at least 3 Triggers on the database and justify it.
- Write at least 3 Cursors on the database and justify it.

DESCRIPTION

PL/SQL is the procedural constructs of SQL which allow us to write an SQL query in procedural way. This helps in executing multiple SQL statements in a single query. This concept is helpful when creating the HISTORY table in database to log the list of events occurred on the table. There could be many more applications of PL/SQL in the web project

EXPECTED OUTCOME

Students should be able to write and execute the PL/SQL and take a snap of the result and prepare a report for the same

CANTEEN MANAGEMENT SYSTEM

CREATING DATABASE :

```sql
create database canteen;
use canteen;
CREATE TABLE RATING (
    USER_ID INT,
    Points NUMERIC,
    menu_id INT,
    ID VARCHAR(25) PRIMARY KEY,
    feedback VARCHAR(50),
    date_time DATE
);
```

```sql
CREATE TABLE MENU (
    MNAME VARCHAR(20),
    ID VARCHAR(25),
    PRICE INT,
    TYPE_ID VARCHAR(25),
    INGREDIENTS VARCHAR(50),
    AVABILITTY_STATUS VARCHAR(100)
);

```

```sql
CREATE TABLE MENU_TYPE (
    DESRIPTION VARCHAR(200),
    ID VARCHAR(25),
    type_name VARCHAR(20)
);

```

```sql
CREATE TABLE CUSTOMER (
    ID VARCHAR(25),
    FIRST_NAME VARCHAR(50),
    USER_EMAIL_ID VARCHAR(70),
    LAST_NAME VARCHAR(20),
    USERNAME VARCHAR(25),
    MPASSWORD VARCHAR(15),
    ACC_STATUS VARCHAR(30),
    USER_CONTACT_NUMBER INT
);

```

```sql
CREATE TABLE PAYMENT (
    ID VARCHAR(25),
    PAYMENT_BY VARCHAR(30),
    ODATE DATE,
    ORDER_ID VARCHAR(20),
    AMOUNT INT
);
```

```sql

CREATE TABLE ORDER_DETAILS (
    ID VARCHAR(25),
    ORDER_ID VARCHAR(20),
    MENU_ID INT,
    AMOUNT INT,
    QUANTITY INT
);
```

```
52
53  CREATE TABLE OORDER (
54      ID VARCHAR(25),
55      CUSTOMER_ID INT,
56      TOTAL_AMOUNT INT,
57      ORDER_STATUS VARCHAR(15)
58  );
59
```

```
60  CREATE TABLE WEBSITE (
61      ID VARCHAR(25),
62      USER_ID INT,
63      CNAME VARCHAR(50),
64      DESCRIPTION VARCHAR(200),
65      CONTACT_INFO INT
66  );
```

```
67
68  CREATE TABLE UUSER (
69      EMAIL_ADD VARCHAR(70),
70      ID VARCHAR(25),
71      CONTACT_NO INT,
72      MPASSWORD VARCHAR(15),
73      FULL_NAME VARCHAR(50),
74      USERNAME VARCHAR(25)
75  );
76
```

DATA INSERTION :

```
insert into RATING values ( 111, 8, 1001, 'AU20B1001', 'GOOD' , '2021-12-01');
insert into RATING values ( 112, 6, 1002, 'AU20B1004', 'OK' , '2021-11-02');
select*from RATING ;
```

```
INSERT into MENU VALUES ( 'CHILLI PANEER' ,'AU20B1001' , '110' ,
'VEG', 'CAPSICUM, ONIONS, VEG', 'AVAILABLE');
INSERT into MENU VALUES ( 'SANDWICH' ,'AU20B1004' , '70' , 'VEG',
'CAPSICUM, ONIONS, BREAD, BUTTER', 'UNAVAILABLE');
SELECT*FROM MENU;
```

```
INSERT INTO MENU_TYPE VALUES ( 'YOU CAN GO FOR THIS DISH IF YOU
ARE SUPER HUNGERY AND WANT TO EAT SOEMTHING SPECIAL',
'AU20B1001', 'VEG');
INSERT INTO MENU_TYPE VALUES ( 'BEST TO HAVE ANYTIME ',
'AU20B1004', 'VEG');
SELECT*FROM MENU_TYPE;
```

```
INSERT INTO CUSTOMER VALUES ('AU20B1001', 'VIDUSHI', 'VID4
GMAIL.COM' , 'YADAV' , 'VID.YA' , 'GGMMAIL65' , 'NO', 996417 );
INSERT INTO CUSTOMER VALUES ('AU20B1004', 'RISHAN',
'RISH64@GMAIL.COM' , 'SHARMA' , 'RISH_SHA' , 'BJI8653@' , 'YES',
974264 );
select * from customer;
```

```
INSERT INTO PAYMENT VALUES ('AU20B1001', 'CASH', '2021-09-04' ,'AF608J', 110);
INSERT INTO PAYMENT VALUES ('AU20B1004', 'paytm', '2021-09-09' ,'GU764P', 70);
select * from payment;

INSERT INTO ORDER_DETAILS  VALUES ('AU20B1001', 'AF608J' , 13685 , 110 ,1 );
INSERT INTO ORDER_DETAILS  VALUES ('AU20B1004', 'GU764P' , 86432, 70 ,3 );
select * from ORDER_DETAILS;

insert into OORDER VALUES ('AU20B1001' ,976438 , 110 ,  '10 MINS' );
insert into OORDER VALUES ('AU20B1004' ,680271 , 240 ,  '2 MINS' );
select * from OORDER;

insert into WEBSITE VALUES ('AU20B1001' , 111 , 'VIDUSHI YADAV' ,
'YOU CAN GO FOR THIS DISH IF YOU ARE SUPER HUNGERY AND WANT
TO EAT SOEMTHING SPECIAL', 9964);
insert into WEBSITE VALUES ('AU20B1004' , 112 , 'RISHAN SHARMA' ,
'BEST TO HAVE ANYTIME ' ,97426);
select * from Website;

INSERT INTO UUSER VALUES ('AU20B1001' , 111 , 99641 , 'GGMMAIL65' ,
'VIDUSHI YADAV' , 'VID.YA' );
INSERT INTO UUSER VALUES ('AU20B1004' , 112 , 974264  , 'BJI8653@'  ,
'RISHAN SHARMA' , 'RISH_SHA' );
select * from Uuser;
```

DATABASE TABLES :

RATING :

USER_ID	Points	menu_id	ID	feedback	date_time
111	8	1001	AU20B1001	GOOD	2021-12-01
112	6	1002	AU20B1004	OK	2021-11-02
NULL	NULL	NULL	NULL	NULL	NULL

MENU :

MNAME	ID	PRICE	TYPE_ID	INGREDIENTS	AVABILITTY_STATUS
CHILLI PANEER	AU20B1001	60	VEG	CAPSICUM, ONIONS, VEG	AVAILABLE
CHILLI PANEER	AU20B1001	60	VEG	CAPSICUM, ONIONS, VEG	AVAILABLE
CHILLI PANEER	AU20B1001	110	VEG	CAPSICUM, ONIONS, VEG	AVAILABLE
SANDWICH	AU20B1004	70	VEG	CAPSICUM, ONIONS, BREAD, BUTTER	UNAVAILABLE

MENU_TYPE :

DESRIPTION	ID	type_name
YOU CAN GO FOR THIS DISH IF YOU ARE SUPER HUNGERY AND WANT TO EAT SOEMTHING SPECIAL	AU20B1001	VEG
BEST TO HAVE ANYTIME	AU20B1004	VEG

CUSTOMER :

ID	FIRST_NAME	USER_EMAIL_ID	LAST_NAME	USERNAME	MPASSWORD	ACC_STATUS	USER_CONTACT_NUMBER
AU20B1001	VIDUSHI	VID4 GMAIL.COM	YADAV	VID.YA	GGMMAIL65	NO	996417
AU20B1004	RISHAN	RISH64@GMAIL.COM	SHARMA	RISH_SHA	BJI8653@	YES	974264

PAYMENT :

ID	PAYMENT_BY	ODATE	ORDER_ID	AMOUNT
AU20B1001	CASH	2021-09-04	AF608J	110
AU20B1004	paytm	2021-09-09	GU764P	70

ORDER_DETAILS :

ID	ORDER_ID	MENU_ID	AMOUNT	QUANTITY
AU20B1001	AF608J	13685	110	1
AU20B1004	GU764P	86432	70	3

OORDER :

	ID	CUSTOMER_ID	TOTAL_AMOUNT	ORDER_STATUS
▶	AU20B1001	976438	110	10 MINS
	AU20B1004	680271	240	2 MINS

WEBSITE :

	ID	USER_ID	CNAME	DESCRIPTION	CONTACT_INFO
▶	AU20B1001	111	VIDUSHI YADAV	YOU CAN GO FOR THIS DISH IF YOU ARE SUPE...	9964
	AU20B1004	112	RISHAN SHARMA	BEST TO HAVE ANYTIME	97426

UUSER :

	EMAIL_ADD	ID	CONTACT_NO	MPASSWORD	FULL_NAME	USERNAME
▶	AU20B1001	111	99641	GGMMAIL65	VIDUSHI YADAV	VID.YA
	AU20B1004	112	974264	BJI8653@	RISHAN SHARMA	RISH_SHA

PROCEDURES

PROCEDURE 1

Create Procedure for payment details of the selected food item

```
131
132     /*Create Procedure for payment details of the selected food item*/
133     delimiter //
134     CREATE PROCEDURE payment_detail()
135     BEGIN
136     SELECT amount from payment where amount > 70;
137     end //
138
139
140     call payment_detail();
141
142
143
```

	amount
▶	110

PROCEDURE 2

Create Procedure for displaying customer name

```
144     /*Create Procedure for displaying customer name */
145
146     delimiter $$
147     CREATE PROCEDURE CUSTOMER_NAME()
148     begin
149     SELECT first_name, last_name from customer;
150     end $$
151     delimiter ;
152
153     call CUSTOMER_NAME();
154
```

first_name	last_name
VIDUSHI	YADAV
RISHAN	SHARMA

PROCEDURE 3

Create Procedure for displaying selected menu item name

```
157    /*Create Procedure for displaying selected menu item name*/
158    delimiter //
159    CREATE PROCEDURE MENU_ITEM_NAME()
160    begin
161    Select * from menu where mname ='Chilli Paneer';
162    end //
163    delimiter ;
164
165    call MENU_ITEM_NAME()
166
167
168
```

	MNAME	ID	PRICE	TYPE_ID	INGREDIENTS	AVABILITTY_STATUS
▶	CHILLI PANEER	AU20B1001	110	VEG	CAPSICUM, ONIONS, VEG	AVAILABLE

TRIGGERS

TRIGGER 1

```
165     /* triggers*/
166     delimiter $$
167     create trigger add_customer before insert on customer for each row
168     begin
169     insert into customer values (45, 'Sandy', 'sandygmail.com', 'Shah','Sshah','sha123','Pending','90876543');
170     end $$
171     delimiter ;
172
```

TRIGGER 2

```
173
174     /*trigger 2*/
175     delimiter $$
176     create trigger MENU_UPDATE before insert on MENU for each row
177     begin
178     insert into MENU values ('PASTA' ,'AU20B1007' , '110' , 'VEG', 'CAPSICUM, ONIONS, VEG, PASTA , SAUCE', 'UNAVAILABLE');
179     end $$
180     delimiter ;
181
```

TRIGGER 3

```
181
182
183     /*trigger 3*/
184     delimiter $$
185     create trigger menu_id before insert on ORDER_DETAILS  for each row
186     begin
187     insert into ORDER_DETAILS  values ('AU20B1001', 'AF608J' , 85385 , 110 ,1 );
188     end $$
189     delimiter ;
190
191
```

CURSORS

CURSOR 1 :

```
194
195     -- Cursor-1 Create a list of every food item on the menu
196
197     DELIMITER $$
198     CREATE PROCEDURE foodList (
199         INOUT fList varchar(4000)
200     )
201     BEGIN
202         DECLARE finished INTEGER DEFAULT 0;
203         DECLARE food varchar(100) DEFAULT "";
204         DECLARE curName
205             CURSOR FOR
206                 SELECT mname FROM menu;
207         DECLARE CONTINUE HANDLER
208             FOR NOT FOUND SET finished = 1;
209         OPEN curName;
210         getfood: LOOP
211             FETCH curName INTO food;
212             IF finished = 1 THEN
213                 LEAVE getfood;
214             END IF;
215             SET fList = CONCAT(food,";",fList);
216         END LOOP getfood;
217         CLOSE curName;
218     END$$
219     DELIMITER ;
220
221     SET @fList = "";
222     CALL foodList(@fList);
```

OUTPUT :

	@fList
▶	SANDWICH;CHILLI PANEER;

CURSOR 2 :

```
226
227     -- Cursor-2 Find most expensive item on the menu
228
229     DELIMITER $$
230     CREATE PROCEDURE maxPrice ()
231     BEGIN
232         DECLARE m_price numeric(20);
233         DECLARE finished INT DEFAULT 0;
234         DECLARE max_price
235             CURSOR FOR
236                 SELECT max(price) FROM menu;
237         DECLARE CONTINUE HANDLER
238             FOR NOT FOUND SET finished = 1;
239         OPEN max_price;
240         getPrice: LOOP
241             FETCH max_price INTO m_price;
242             IF finished = 1 THEN
243                 LEAVE getPrice;
244             END IF;
245             SELECT CONCAT(m_price);
246         END LOOP getPrice;
247         CLOSE max_price;
248     END$$
249     DELIMITER ;
250
251     call maxPrice();
```

	CONCAT(m_price)
▶	110

CURSOR 3 :

```
255    -- Cursor-3 Get feedback given by a specific person
256
257    DELIMITER $$
258    CREATE PROCEDURE get_feedback (in custname varchar(50))
259    BEGIN
260        DECLARE cname varchar(50);
261        DECLARE rfeedback varchar(50);
262        DECLARE finished INT DEFAULT 0;
263        DECLARE cust_name
264            CURSOR FOR
265                select c.first_name, r.feedback from customer c, rating r where c.id = r.id and c.first_name = custname;
266        DECLARE CONTINUE HANDLER
267            FOR NOT FOUND SET finished = 1;
268        OPEN cust_name;
269        getName: LOOP
270            FETCH cust_name INTO cname, rfeedback;
271            IF finished = 1 THEN
272                LEAVE getName;
273            END IF;
274            SELECT CONCAT(cname, ";", rfeedback);
275        END LOOP getName;
276        CLOSE cust_name;
277    END$$
278    DELIMITER ;
279
280    call get_feedback ('Vidushi');
```

	CONCAT(cname, ";", rfeedback)
▶	VIDUSHI;GOOD

CONCLUSION

- SQL is the standard database language and PL/SQL is strongly integrated with SQL. PL/SQL supports both static and dynamic SQL. Static SQL supports DML operations and transaction control from PL/SQL block. In Dynamic SQL, SQL allows embedding DDL statements in PL/SQL blocks.

- PL/SQL allows sending an entire block of statements to the database at one time. This reduces network traffic and provides high performance for the applications.

- PL/SQL gives high productivity to programmers as it can query, transform, and update data in a database.

- PL/SQL saves time on design and debugging by strong features, such as exception handling, encapsulation, data hiding, and object-oriented data types.

- Applications written in PL/SQL are fully portable.

- PL/SQL provides high security level.

- PL/SQL provides access to predefined SQL packages.

- PL/SQL provides support for Object-Oriented Programming.

- PL/SQL provides support for developing Web Applications and Server Pages