1. Insert Record using

Anonymous Block

Create a PL/SQL block to insert a new record into the Department table. Fetch the maximum department id from the Department table and add 10 to it; take this value for department id; 'TESTING' is the value for department name and CHN-102 is the value for Location ID.

Note: Use '/' to terminate your query before compilation and evaluation

Table name : Department

Column name	Data type	Constraints
DEPARTMENT_ID	NUMBER(5)	PK
DEPARTMENT_NAME	VARCHAR2(25)	NOT NULL
LOCATION_ID	VARCHAR2(15)	

Sample Output:

```
DEPARTMENT_ID   DEPARTMENT_NAME          LOCATION_ID
--------------- ------------------------------ ----------------
XXXX            TESTING                  CHN-102
```

1. Answer

==========

2. Update Location

Create a PL/SQL block to update the location ID for an existing department, which has location ID preceded with 'HQ' as 'HQ-BLR-101'.

Note: Use '/' to terminate your query before compilation and evaluation

Table name : **Department**

Column name	Data type	Constraints
DEPARTMENT_ID	NUMBER(5)	PK
DEPARTMENT_NAME	VARCHAR2(25)	NOT NULL
LOCATION_ID	VARCHAR2(15)	

Sample Output:

```
DEPARTMENT_ID       DEPARTMENT_NAME    LOCATION_ID
------------------- -------------------------- ----------------------
  xxxx                xxxxx                HQ-BLR-101
```

==========

3. Area of a Circle

Write a PL/SQL block to calculate the area of a circle for the radius ranging from 3 to 7 . Store the radius and corresponding area into the Circle table.

Circle :

Radius Number(5)

AreaNumber(7,2)

Assume that the circle table has been already created.

Note: Use '/' to terminate your query before compilation and evaluation.

3. Answer

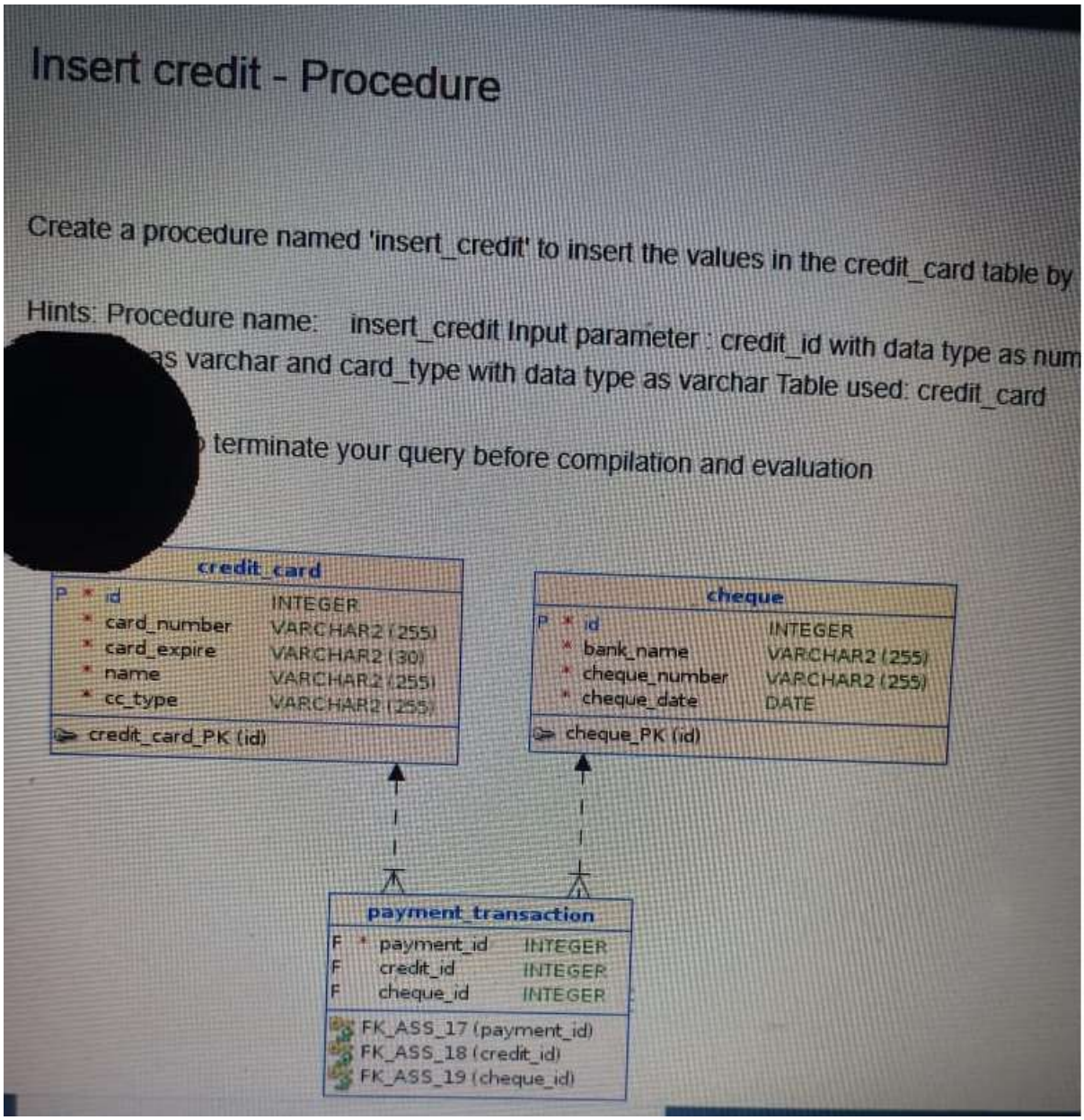
==========

4. Insert credit - Procedure

Create a procedure named 'insert_credit' to insert the values in the credit_card table by passing 5 inputs as parameters.

Hints: Procedure name: insert_credit Input parameter : credit_id with data type as number,credit_card_number with data type as varchar,credit_card_expire with data type as varchar,holder_name with data type as varchar and card_type with data type as varchar Table used: credit_card

Note: Use '/' to terminate your query before compilation and evaluation



==========

5. Select city - Procedure

Create a procedure named 'select_city' which accepts one input parameter

user_id of type number and one output parameter city_details of type varchar. This procedure is used to display the city_details of user.If the user is from Bangalore then display the city_details as 'User is from Bangalore',or if the user is from Chennai then display the city_details as 'User is from Chennai', else display the city_details as 'User is from other cities'.

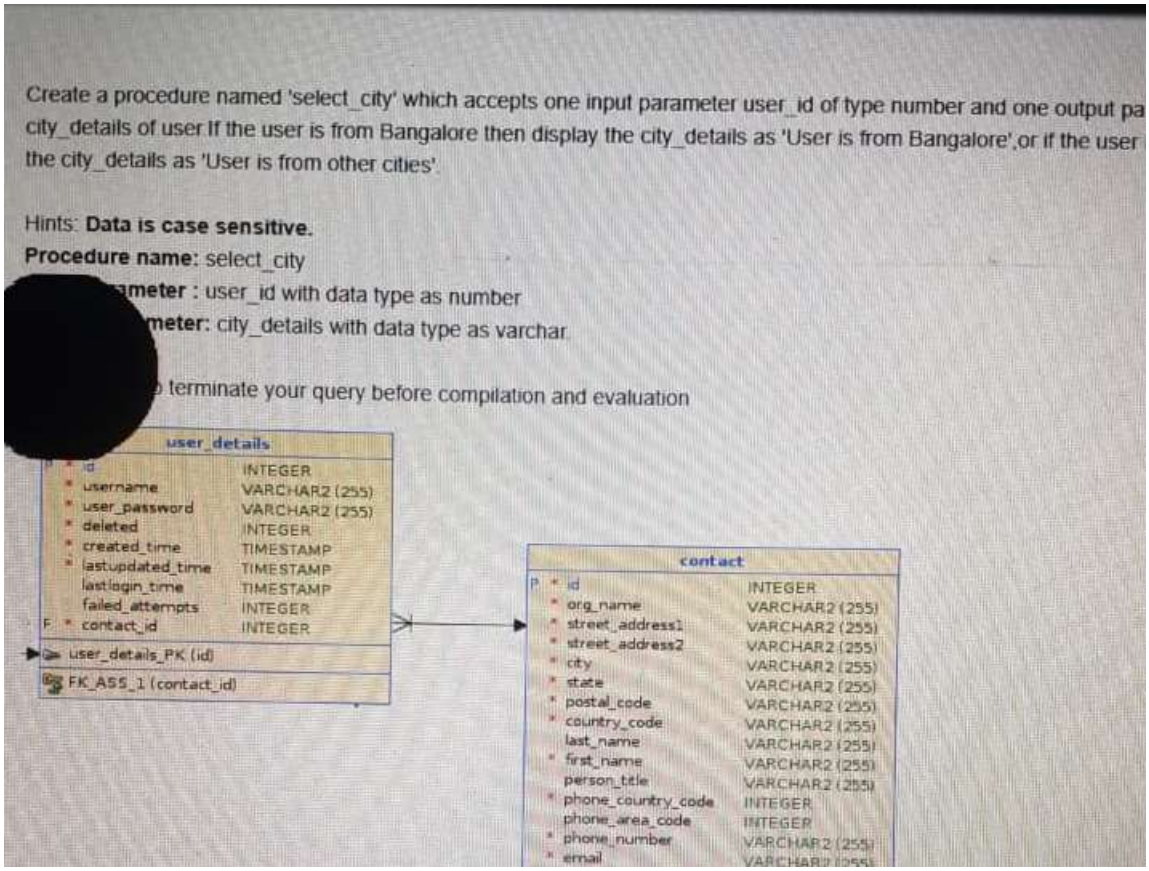
Hints: Data is case sensitive.

Procedure name: select_city

Input parameter : user_id with data type as number

Output parameter: city_details with data type as varchar.

Note: Use '/' to terminate your query before compilation and evaluation



5. Answer

==========

6. Insert a Record - Triggers

Create a PL/SQL Trigger to display the message “NEW EMPLOYEE DETAILS INSERTED”, whenever a new record is inserted into Employee table.

Column name	Data type	Constraints
EMP_ID	NUMBER(5)	PK
EMP_NAME	VARCHAR2(25)	NOT NULL
SALARY	NUMBER(10,2)	

Note: Use '/' to terminate your query before compilation and evaluation

==========

7. Package with a Procedure to update salary

Create a PL/SQL Package with Procedure in it. Procedure will take designation and incentive as input and update the employee salary by adding the incentive for the given designation. Display the number of employee records that have got updated, e.g. ‘3 employee record(s) are updated’.

Note: Use '/' to terminate your query before compilation and evaluation

Employee:

Column name	Data type	Constraints
EMP_ID	NUMBER(5)	PK
EMP_NAME	VARCHAR2(25)	NOT NULL
SALARY	NUMBER(10,2)	
DESIGNATION	VARCHAR2(25)	

EMP_ID	EMP_NAME	SALARY	DESIGNATION
101	Mathew	45000	PROGRAMMER
102	Sam	54000	MANAGER

103	John	35000	TEAM LEAD
104	James	48000	PROGRAMMER
105	Josh	25000	TESTER

Functional Requirements:

Package name as EMP_DESIGNATION, and

Procedure signature:

EMP_DETAILS(design employee.designation%TYPE, incentive number)

7 . Answer

==========

8. Display department names using Cursors

Create a PL/SQL block to display all the department names from the Department table using cursors. The department names should be displayed in ascending order.

Column name	Data type	Constraints
DEPARTMENT_ID	NUMBER(5)	PK
DEPARTMENT_NAME	VARCHAR2(25)	NOT NULL
LOCATION_ID	VARCHAR2(15)	

Sample Output:

Department Names are :

ADMIN

DEVELOPMENT

Note: Use '/' to terminate your query before compilation and evaluation

8 . Answer

==========

9. Procedure with Exception Handling

Create a PL/SQL Procedure to insert employee details into Employee table. Before inserting, check whether the employee age is eligible or not. Employee age should be 18 or greater. Values are passed as argument to the procedure.

If age valid, insert employee record into table and print the message "Age valid - Record inserted", else print the message "Age invalid - Record not inserted" by raising an exception.

Table: EMPLOYEE

Column name	Data type	Constraints
EMP_ID	NUMBER(5)	PK
EMP_NAME	VARCHAR2(25)	NOT NULL
AGE	NUMBER(3)	

Note: Use '/' to terminate your query before compilation and evaluation

Functional Requirement:

PROCEDURE CHECK_AGE_ELIGIBILITY(

v_id IN EMPLOYEE.EMPID%TYPE,

v_name IN EMPLOYEE.EMPNAME%TYPE,

v_age IN EMPLOYEE.AGE%TYPE)

Sample Input 1 :

CHECK_AGE_ELIGIBILITY(103, 'Robert', 24) ;

Sample Output 1:

Age valid - Record inserted

Sample Input 2:

CHECK_AGE_ELIGIBILITY(104,'Riya', 4);

Sample Output 2:

Age invalid - Record not inserted

==========