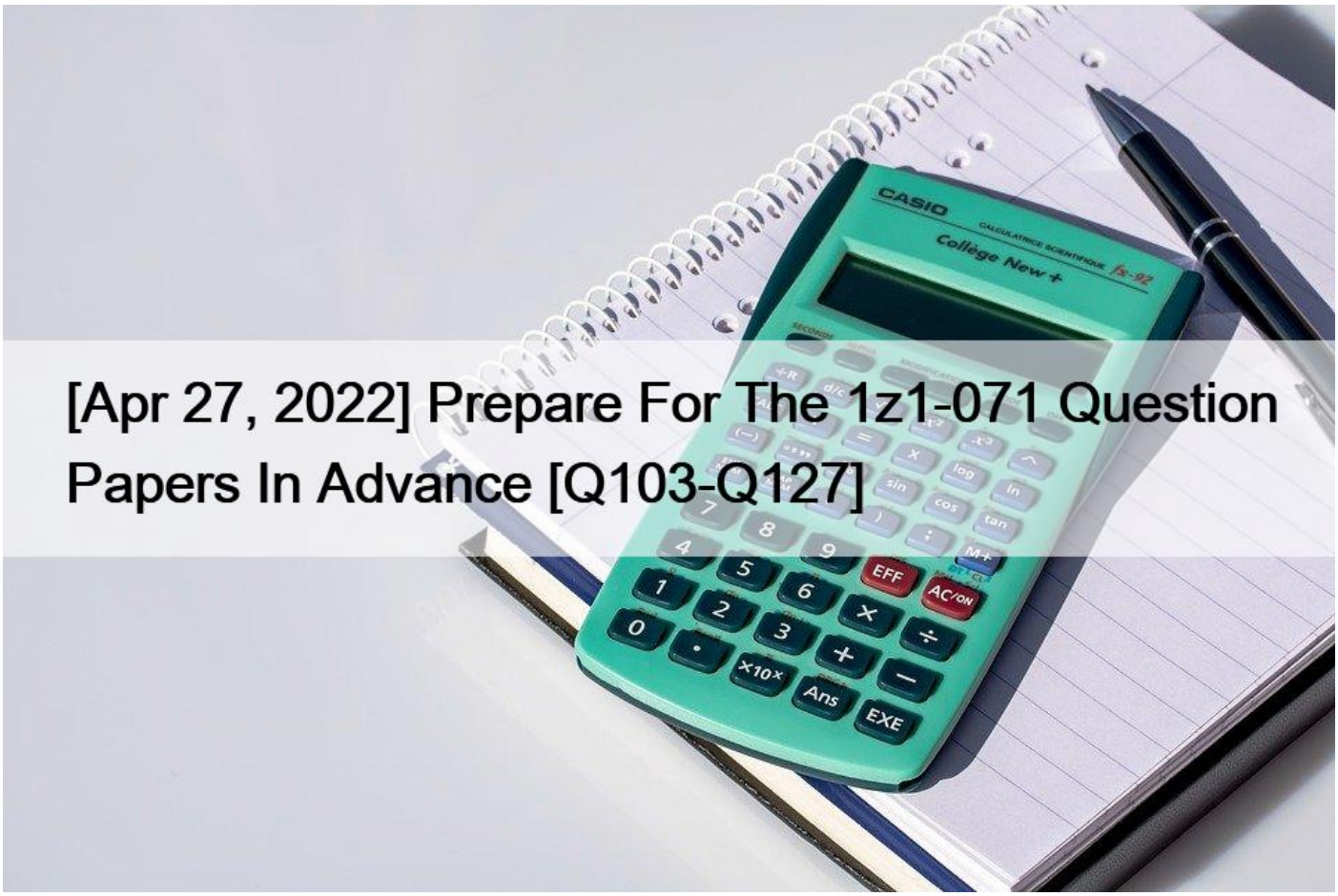
[Apr 27, 2022 Prepare For The 1z1-071 Question Papers In Advance [Q103-Q127



[Apr 27, 2022] Prepare For The 1z1-071 Question Papers In Advance
1z1-071 PDF Dumps Real 2022 Recently Updated Questions

NO.103 View the Exhibit and examine the structure of the PROMOTIONS table.

Evaluate the following SQL statement:

Which statement is true regarding the outcome of the above query?
* It produces an error because subqueries cannot be used with the CASE expression.
* It shows COST_REMARK for all the promos in the promo category ‘TV’.
* It shows COST_REMARK for all the promos in the table.
* It produces an error because the subquery gives an error.

NO.104 Examine the structure of the ORDERS table: (Choose the best answer.)

NAME	NULL	TYPE
ORDER_ID	NOT NULL	NUMBER (12)
ORDER_DATE	NOT NULL	TIMESTAMP(6)
CUSTOMERS_ID	NOT NULL	NUMBER(6)
ORDER_STATUS		NUMBER(2)
ORDER_TOTAL		NUMBER(8, 2)

You want to find the total value of all the orders for each year and issue this command:

SQL> SELECT TO_CHAR(order_date,’rr’), SUM(order_total) FROM orders

GROUP BY TO_CHAR(order_date, ‘yyyy’);

Which statement is true regarding the result?
* It executes successfully but does not give the correct output.
* It executes successfully and gives the correct output.
* It returns an error because the TO_CHAR function is not valid.
* It return an error because the datatype conversion in the SELECT list does not match the data type conversion in the GROUP BY clause.

NO.105 Evaluate the following SQL statements that are issued in the given order: CREATE TABLE emp

(emp_no NUMBER(2) CONSTRAINT emp_emp_no_pk PRIMARY KEY,

enameVARCHAR2(15),

salary NUMBER(8,2),

mgr_no NUMBER(2) CONSTRAINT emp_mgr_fk REFERENCES emp);

ALTER TABLE emp

DISABLE CONSTRAINT emp_emp_no_pk CASCADE;

ALTER TABLE emp

ENABLE CONSTRAINT emp_emp_no_pk;

What would be the status of the foreign key EMP_MGR_FK?
* It would be automatically enabled and deferred.
* It would be automatically enabled and immediate.
* It would remain disabled and has to be enabled manually using the ALTER TABLE command.
* It would remain disabled and can be enabled only by dropping the foreign key constraint and re-creating it.

NO.106 Evaluate these commands which execute successfully:

```
CREATE SEQUENCE ord_seq
  INCREMENT BY 1
  START WITH 1
  MAXVALUE 100000
  CYCLE
  CACHE 5000;

CREATE TABLE ord_items
  ord_no      NUMBER(4) DEFAULT ord_seq.NEXTVAL NOT NULL,
  item_no     NUMBER(3),
  qty         NUMBER(3),
  expiry_date DATE,
  CONSTRAINT it_pk  PRIMARY KEY (ord_no, item_no),
  CONSTRAINT ord_fk FOREIGN KEY (ord_no) REFERENCES orders (ord_no));
```


Which two statements are true about the ORD_ITEMStable and the ORD_SEQsequence? (Choose two.)
* Sequence ORD_SEQcycles back to 1 after every 5000 numbers and can cycle 20 times.
* Any user inserting rows into table ORD_ITEMSmust have been granted access to sequence ORD_SEQ.
* Column ORD_NOgets the next number from sequence ORD_SEQwhenever a row is inserted into ORD_ITEMSand no explicit value is given for ORD_NO.
* If sequence ORD_SEQis dropped then the default value for column ORD_NOwill be NULL for rows inserted into ORD_ITEMS.
* Sequence ORD_SEQis guaranteed not to generate duplicate numbers.

NO.107 Examine the structure of the BOOKS_TRANSACTIONS table:

```
Name                      Null?          Type
------------------------------------------------------------------------------
TRANSACTION_ID            NOT NULL       VARCHAR2 (6)
TRANSACTION_TYPE                         VARCHAR2 (3)
BORROWED_DATE                            DATE
DUE_DATE                                 DATE
BOOK_ID                                  VARCHAR2 (6)
MEMBER_ID                                VARHCAR2 (6)
```


Examine the SQL statement:

```
SQL> SELECT * FROM books_transactions WHERE borrowed_date<SYSDATE
AND transaction_type= 'RM' OR MEMBER_ID IN ('A101', 'A102');
```

Which statement is true about the outcome?
* It displays details only for members who have borrowed before today with RM as TRANSACTION_TYPE.
* It displays details for members who have borrowed before today’s date with either RM as TRANSACTION_TYPE or MEMBER_ID as A101 and A102.
* It displays details for only members A101 and A102 who have borrowed before today with RM TRANSACTION_TYPE.
* It displays details for members who have borrowed before today with RM as TRANSACTION_TYPE and the details for members A101 or A102.

NO.108 Examine the data in the CUST_NAME column of the CUSTOMERS table.

```
CUST_NAME
--------------------
Lex De Haan
Renske Ladwig
Jose Manuel Urman
Jason Mallin
```

You want to extract only those customer names that have three names and display the * symbol in place of the first name as follows:

```
CUST NAME
--------------------
*** De Haan
*** Manuel Urman
```

Which two queries give the required output?
* SELECT LPAD(SUBSTR(cust_name, INSTR(cust_name, ‘ ‘)),LENGTH(cust_name),’*’) “CUST NAME”FROM customersWHERE INSTR(cust_name, ‘ ‘,1,2)<>0;
* SELECT LPAD(SUBSTR(cust_name, INSTR(cust_name, ‘ ‘)),LENGTH(cust_name),’*’) “CUST NAME”FROM customersWHERE INSTR(cust_name, ‘ ‘,-1,2)<>0;
* SELECT LPAD(SUBSTR(cust_name ‘ ‘)),LENGTH(cust_name) – INSTR(cust_name, ‘ ‘), ‘*’) “CUST NAME”FROM customersWHERE INSTR(cust_name, ‘ ‘,1,-2)<>0;
* SELECT LPAD(SUBSTR(cust_name ‘ ‘)),LENGTH(cust_name) – INSTR(cust_name, ‘ ‘), ‘*’) “CUST NAME”FROM customersWHERE INSTR(cust_name, ‘ ‘,1,2)<>0;

NO.109 Examine the description of the EMPLOYEES table:

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(3)
FIRST_NAME		VARCHAR2(15)
LAST_NAME	NOT NULL	VARCHAR2(15)
SALARY		NUMBER(6,2)

Which statement will execute successfully, returning distinct employees with non-null first names?
* SELECT DISTINCT * FROM employees WHERE first_ name IS NOT NULL;
* SELECT first_ name, DISTNCT last_ name FROM employees WHERE first_ name IS NOT NULL;
* SELECT Distinct * FROM employees WHERE first_ name < > NULL;
* SELECT first_ name, DISTINCT last_ name FROM employees WHERE first_ name < > NULL;

NO.110 Examine this partial statement:

SELECT ename, sal,comm FROM emp

Now examine this output:

ENAME		SAL	COMM
MARTIN		1250	1400
WARD		1250	500
ALIEN		1600	300
TURNER		1500	0
ADAMS		1100	
BLARE		2850	
CLARR		2450	
FORD		3000	
JAMES		950	
JONES		2975	
RING		5000	
MILLER	1300		
SCOTT		3000	
SMITH		800	

WHICH ORDER BY clause will generate the displayed output?
* ORDER BY NVL(enam,0) DESC, ename
* ORDER BY NVL(comm,0) ASC NULLS FIRST, ename
* ORDER BY NVL(comm,0) ASC NULLS LAST, ename
* ORDER BY comm DESC NULLS LAST, ename

NO.111 Examine the structure of the EMPLOYEES table. (Choose the best answer.)

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

You must display the details of employees who have manager with MANAGER_ID 100, who were hired in the past 6 months and who have salaries greater than 10000.
* SELECT last_name, hire_date, salaryFROM employeesWHERE salary > 10000UNION ALL SELECT last_name, hire_date, salaryFROM employeesWHERE manager_ID = (SELECT employee_id FROM employees WHERE employee_id

100)INETRSECTSELECT last_name, hire_date, salaryFROM employees WHERE

hire_date > SYSDATE- 180;
* SELECT last_name, hire_date, salaryFROM employeesWHERE manager_id

(SELECT employee_id FROM employees WHERE employee_id = 100)UNION

ALL(SELECT last_name, hire_date, salaryFROM employeesWHERE hire_date >

SYSDATE -180INTERSECTSELECT last_name, hire_date, salaryFROM

employeesWHERE salary > 10000);
* SELECT last_name, hire_date, salaryFROM employeesWHERE manager_id

(SELECT employee_id FROM employees WHERE employee_id = ‘100’)UNIONSELECT last_name, hire_date, salaryFROM employeesWHERE hire_date > SYSDATE –

1 80INTERSECTSELECT last_name, hire_date, salaryFROM employeesWHERE salary >

1 0000;
* (SELECT last_name, hire_date, salaryFROM employeesWHERE salary > 10000UNION ALLSELECT last_name, hire_date, salaryFROM employeesWHERE manager_ID = (SELECT employee_id FROM employees WHERE employee_id = 100))UNIONSELECT last_name, hire_date, salaryFROM employeesWHERE hire_date > SYSDATE -180;

NO.112 Examine the description of the EMPLOYEES table:

```
Name            Null?    Type
--------------- -------- ----------------------------
EMP_ID          NOT NULL NUMBER
EMP_NAME                 VARCHAR2(10)
DEPT_ID                  NUMBER(2)
SALARY                   NUMBER(8,2)
JOIN_DATE                DATE

NLS_DATE_FORMAT is set to DD-MON-YY.
```

Which query requires explicit data type conversion?
* SELECT join_date FROM employees WHERE join_date > ’10-02-2018′;
* SELECT salary + ‘120.50’ FROM employees;
* SELECT SUBSTR(join_date, 1, 2) – 10 FROM employees;
* SELECT join_date + ’20’ FROM employees;
* SELECT join_date || ‘ ‘ || salary FROM employees;

Explanation

NO.113 View the exhibits and examine the structures of the COSTS and PROMOTIONS tables.

Table COSTS

Name	Null?	Type
PROD_ID	NOT NULL	NUMBER
TIME_ID	NOT NULL	DATE
PROMO_ID	NOT NULL	NUMBER
CHANNEL_ID	NOT NULL	NUMBER
UNIT_COST	NOT NULL	NUMBER (10,2)
UNIT_PRICE	NOT NULL	NUMBER (10,2)

Table PROMOTIONS

Name	Null?	Type
PROMO_ID	NOT NULL	NUMBER(6)
PROMO_NAME	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY_ID	NOT NULL	NUMBER
PROMO_CATEGORY	NOT NULL	VARCHAR2(30)
PROMO_CATEGORY_ID	NOT NULL	NUMBER
PROMO_COST	NOT NULL	NUMBER(10,2)
PROMO_BEGIN_DATE	NOT NULL	DATE
PROMO_END_DATE	NOT NULL	DATE

Evaluate the following SQL statement:

```
SQL> SELECT prod_id
     FROM costs
     WHERE promo_id IN  (SELECT promo_id FROM promotions
                         WHERE promo_cost < ALL
                          (SELECT MAX(promo_cost) FROM promotions
                           GROUP BY (promo_end_date - promo_begin_date)));
```

What would be the outcome of the above SQL statement?

* It displays prod IDs in the promo with the lowest cost.
* It displays prod IDs in the promos with the lowest cost in the same time interval.
* It displays prod IDs in the promos with the highest cost in the same time interval.
* It displays prod IDs in the promos which cost less than the highest cost in the same time interval.

NO.114 View the exhibit and examine the structure of the SALES, CUSTOMERS, PRODUCTS and TIMES tables.

The PROD_ID column is the foreign key in the SALES table referencing the PRODUCTS table.

The CUST_ID and TIME_ID columns are also foreign keys in the SALES table referencing the CUSTOMERS and TIMES tables, respectively.

Examine this command:

CREATE TABLE new_sales (prod_id, cust_id, order_date DEFAULT SYSDATE)

AS

SELECT prod_id, cust_id, time_id

FROM sales;

Which statement is true?

* The NEW_SALES table would get created and all the FOREIGN KEY constraints defined on the selected columns from the SALES table would be created on the corresponding columns in the NEW_SALES table.
* The NEW_SALES table would not get created because the column names in the CREATE TABLE command and the SELECT clause do not match.
* The NEW_SALES table would not get created because the DEFAULT value cannot be specified in the column definition.
* The NEW_SALES table would get created and all the NOT NULL constraints defined on the selected columns from the SALES table would be created on the corresponding columns in the NEW_SALES table.

NO.115 Examine the structure of the EMPLOYEES table. (Choose two.)

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

You must display the maximum and minimum salaries of employees hired 1 year ago.

Which two statements would provide the correct output?
* SELECT MIN(Salary) minsal, MAX(salary) maxsalFROM employeesWHERE hire_date

< SYSDATE-365GROUP BY MIN(salary), MAX(salary);
* SELECT minsal, maxsalFROM (SELECT MIN(salary) minsal, MAX(salary) maxsal FROM employeesWHERE hire_date < SYSDATE-365)GROUP BY maxsal, minsal;
* SELECT minsal, maxsalFROM (SELECT MIN(salary) minsal, MAX(salary) maxsal FROM employeesWHERE hire_date < SYSDATE-365GROUP BY MIN(salary), MAX(salary);
* SELECT MIN(Salary), MAX(salary)FROM (SELECT salary FROM employeesWHERE hire_date < SYSDATE-365);

NO.116 Examine the description of the PRODUCTS table:

Name	Null?	Type
PRODUCT_ID	NOT NULL	NUMBER(2)
PRODUCT_NAME		VARCHAR2(10)
UNIT_PRICE		NUMBER(3)
SURCHARGE		VARCHAR2(2)
EXPIRY_DATE		DATE
DELIVERY_DATE		DATE

Which three queries use valid expressions? (Choose three.)

* SELECT product_id, (expiry_date – delivery_date) * 2 FROM products;
* SELECT product_id, unit_price | | 5 “Discount”, unit_price + surcharge – discount FROM products;
* SELECT product_id, unit_price, 5 “Discount”, unit_price + surcharge – discount FROM products;
* SELECT product_id, unit_price, unit_price + surcharge FROM products;
* SELECT product_id, (unit_price * 0.15 / (4.75 + 552.25)) FROM products;
* SELECT product_id, expiry_date * 2 FROM products;

NO.117 Examine this partial command:

```
CREATE TABLE cust (
  cust_id NUMBER(2),
  credit_limit NUMBER(10)
)
ORGANIZATION EXTERNAL
```

Which two clauses are required for this command to execute successfully?
* the LOCATIONclause
* the access driver TYPEclause
* the REJECT LIMITclause
* the DEFAULT DIRECTORYclause
* the ACCESS PARAMETERSclause

NO.118 See the Exhibit and examine the structure of the PROMOTIONS table:

Table PROMOTIONS

Name	Null?	Type
PROMO_ID	NOT NULL	NUMBER(6)
PROMO_NAME	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY	NOT NULL	VARCHAR2(30)
PROMO_SUBCATEGORY_ID	NOT NULL	NUMBER
PROMO_CATEGORY	NOT NULL	VARCHAR2(30)
PROMO_CATEGORY_ID	NOT NULL	NUMBER
PROMO_COST	NOT NULL	NUMBER(10,2)
PROMO_BEGIN_DATE	NOT NULL	DATE
PROMO_END_DATE	NOT NULL	DATE

Using the PROMOTIONS table,

you need to find out the average cost for all promos in the range $0-2000 and $2000-5000 in

category A.

You issue the following SQL statements:

```
SQL>SELECT AVG(CASE
                    WHEN promo_cost BETWEEN 0 AND 2000 AND promo_category='A'
                         THEN promo_cost
                    ELSE null END) "CAT_2000A",
               AVG(CASE
                    WHEN promo_cost BETWEEN 2001 AND 5000 AND promo_category='A'
                         THEN promo_cost
                    ELSE null END) "CAT_5000A"
          FROM promotions;
```


What would be the outcome?

* It generates an error because multiple conditions cannot be specified for the WHEN clause.
* It executes successfully and gives the required result.
* It generates an error because CASE cannot be used with group functions.
* It generates an error because NULL cannot be specified as a return value.

CASE Expression

Facilitates conditional inquiries by doing the work of an IF-THEN-ELSE statement:

CASE expr WHEN comparison_expr1 THEN return_expr1

[WHEN comparison_expr2 THEN return_expr2

WHEN comparison_exprn THEN return_exprn

ELSE else_expr]

END

NO.119 View the exhibit and examine the structures of the EMPLOYEES and DEPARTMENTS tables.

EMPLOYEES

NameNull?Type

——————- ————-

EMPLOYEE_IDNOT NULLNUMBER(6)

FIRST_NAMEVARCHAR2(20)

LAST_NAMENOT NULLVARCHAR2(25)

HIRE_DATENOT NULLDATE

JOB_IDNOT NULLVARCHAR2(10)

SALARYNUMBER(10,2)

COMMISSIONNUMBER(6,2)

MANAGER_IDNUMBER(6)

DEPARTMENT_IDNUMBER(4)

DEPARTMENTS

NameNull?Type

———————- ————-

DEPARTMENT_IDNOT NULLNUMBER(4)

DEPARTMENT_NAMENOT NULLVARCHAR2(30)

MANAGER_IDNUMBER(6)

LOCATION_IDNUMBER(4)

You want to update EMPLOYEES table as follows:

You issue the following command:

SQL> UPDATE employees

SET department_id

(SELECT department_id

FROM departments

WHERE location_id = 2100),

(salary, commission)

(SELECT 1.1*AVG(salary), 1.5*AVG(commission)

FROM employees, departments

WHERE departments.location_id IN(2900, 2700, 2100))

WHERE department_id IN

(SELECT department_id

FROM departments

WHERE location_id = 2900

OR location_id = 2700;

What is outcome?
* It generates an error because multiple columns (SALARY, COMMISSION) cannot be specified together in an UPDATE statement.
* It generates an error because a subquery cannot have a join condition in a UPDATE statement.
* It executes successfully and gives the desired update
* It executes successfully but does not give the desired update

NO.120 View the Exhibit and examine the structure of the PORDUCT_INFORMATION table.

(Choose the best answer.)

PRODUCT_ID column is the primary key.

You create an index using this command:

SQL > CREATE INDEX upper_name_idx

ON product_information(UPPER(product_name));

No other indexes exist on the PRODUCT_INFORMATION table.

Which query would use the UPPER_NAME_IDX index?
* SELECT product_id, UPPER(product_name)FROM product_informationWHERE

UPPER(product_name) = ‘LASERPRO’ OR list_price > 1000;
* SELECT UPPER(product_name)FROM product_information;
* SELECT UPPER(product_name)FROM product_informationWHERE product_id = 2254;
* SELECT product_idFROM product_informationWHERE UPPER(product_name) IN (‘LASERPRO’, ‘CABLE’);

NO.121 Examine the commands used to create DEPARTMENT_DETAILS and

COURSE_DETAILS tables:

```
SQL>CREATE TABLE DEPARTMENT_DETAILS
(DEPARTMENT_ID NUMBER PRIMARY KEY,
DEPARTMENT_NAME   VARCHAR2(50),
HOD               VARCHAR2(50));
SQL>CREATE TABLE COURSE_DETAILS
(COURSE_ID        NUMBER PRIMARY KEY,
COURSE_NAME       VARCHAR2(50),
DEPARTMENT_ID     NUMBER REFERENCES DEPARTMENT_DETAILS (DEPARTMENT_ID));
```


You want to generate a list of all department IDs that do not exist in the COURSE_DETAILS table.

You execute the SQL statement:

```
SQL> SELECT d.department_id FROM course_details c INNER JOIN
department_details d ON c.department_id<>d.department_id;
```

What is the outcome?
* It fails because the join type used is incorrect.
* It executes successfully and displays the required list.
* It executes successfully but displays an incorrect list.
* It fails because the ON clause condition is not valid.

NO.122 Examine these statements which execute successfully:

ALTER SESSION SET NLS_DATE_FORMAT = ‘DD-MON-YYYY HH24 MI: SS’

ALTER SESSION SET TIME_ ZONE = ‘-5:00’;

SELECT DBTIMEZONE, SYSDATE FROM DUAL

Examine the result:

```
DBTIMEZONE   SYSDATE
-----------  ---------------------
+00.00       11-JUL-2019 11:00:00
```

If LOCALTIMESTAMP was selected at the same time what would it return?
* 11-JUL-2019 6,00,00,00000000 AM – 05:00
* 11-JUL-2019 11,00,00,00000000 AM
* 11-JUL-2019 6,00,00,000000 AM
* 11-JUL-2019 11,00,00,000000AM -05:00

NO.123 You create a table by using this command:

CREATE TABLE rate_list (rate NUMBER(6,2));

Which two are true about executing statements? (Choose two.)
* INSERT INTO rate_list VALUES (-10)produces an error.
* INSERT INTO rate_list VALUES (87654.556)inserts the value as 87654.6.
* INSERT INTO rate_list VALUES (0.551)inserts the value as .55.
* INSERT INTO rate_list VALUES (-99.99)inserts the value as 99.99.
* INSERT INTO rate_list VALUES (0.999) produces an error.
* INSERT INTO rate_list VALUES (-.9)inserts the value as -.9.

NO.124 In the customers table, the CUST_CITY column contains the value ‘Paris’ for the CUST_FIRST_NAME ‘Abigail’.

Evaluate the following query:

```
SQL> SELECT INITCAP(cust_first_name || ' ' ||
           UPPER(SUBSTR(cust_city,-LENGTH(cust_city),2)))
     FROM customers
     WHERE cust_first_name = 'Abigail';
```

What would be the outcome?
* Abigail PA
* Abigail Pa
* Abigail IS
* An error message

NO.125 Examine the commands used to create DEPARTMENT_DETAILS and COURSE_DETAILS tables:

```
SQL>CREATE TABLE DEPARTMENT_DETAILS
(DEPARTMENT_ID NUMBER PRIMARY KEY,
DEPARTMENT_NAME   VARCHAR2(50),
HOD               VARCHAR2(50));
SQL>CREATE TABLE COURSE_DETAILS
(COURSE_ID        NUMBER PRIMARY KEY,
COURSE_NAME       VARCHAR2(50),
DEPARTMENT_ID     NUMBER REFERENCES DEPARTMENT_DETAILS (DEPARTMENT_ID));
```


You want to generate a list of all department IDs that do not exist in the COURSE_DETAILS table.

You execute the SQL statement:

```
SQL> SELECT d.department_id FROM course_details c INNER JOIN
department_details d ON c.department_id<>d.department_id;
```

What is the outcome?
* It fails because the join type used is incorrect.
* It executes successfully and displays the required list.
* It executes successfully but displays an incorrect list.
* It fails because the ON clause condition is not valid.

NO.126 View the Exhibit and examine the details of PRODUCT_INFORMATIONtable.

PRODUCT_NAME CATEGORY_ID SUPPLIER_ID

Inkjet C/8/HQ 12 102094

Inkjet C/4 12 102090

LaserPro 600/6/BW 12 102087

LaserPro 1200/8/BW 12 102099

Inkjet B/6 12 102096

Industrial 700/ID 12 102086

Industrial 600/DQ 12 102088

Compact 400/LQ 12 102087

Compact 400/DQ 12 102088

HD 12GB /R 13 102090

HD 10GB /I 13 102071

HD 12GB @7200 /SE 13 102057

HD 18.2GB @10000 /E 13 102078

HD 18.2GB @10000 /I 13 102050

HD 18GB /SE 13 102083

HD 6GB /I 13 102072

HD 8.2GB@5400 13 102093

You have the requirement to display PRODUCT_NAMEfrom the table where the CATEGORY_IDcolumn has values 12or 13, and the SUPPLIER_IDcolumn has the value 102088. You executed the following SQL statement:

SELECT product_name

FROM product_information

WHERE (category_id = 12 AND category_id = 13) AND supplier_id = 102088; Which statement is true regarding the execution of the query?
* It would not execute because the same column has been used in both sides of the ANDlogical operator to form the condition.
* It would not execute because the entire WHEREclause condition is not enclosed within the parentheses.
* It would execute and the output would display the desired result.
* It would execute but the output would return no rows.

NO.127 Which two statements are true regarding a SAVEPOINT? (Choose two.)
* A SAVEPOINTdoes not issue a COMMIT
* Only one SAVEPOINTmay be issued in a transaction
* Rolling back to a SAVEPOINT can undo a TRUNCATEstatement
* Rolling back to a SAVEPOINTcan undo a CREATE INDEX statement
* Rolling back to a SAVEPOINTcan undo a DELETEstatement