ORACLE

PRACTICA 1

1. **Which statement reports on unique JOB\_ID values from the EMPLOYEES table? (Choose all that apply.)**

A. SELECT JOB\_ID FROM EMPLOYEES;

B. SELECT UNIQUE JOB\_ID FROM EMPLOYEES;

C. SELECT DISTINCT JOB\_ID, EMPLOYEE\_ID FROM EMPLOYEES;

D. SELECT DISTINCT JOB\_ID FROM EMPLOYEES;

**✓ D.** Unique JOB\_ID values are projected from the EMPLOYEES table by applying the

DISTINCT keyword to just the JOB\_ID column.

**A, B,** and **C** are eliminated since **A** returns an unrestricted list of JOB\_ID values including duplicates; **B** makes use of the UNIQUE keyword in the incorrect context; and **C** selects the distinct combination of JOB\_ID and EMPLOYEE\_ID values. This has the effect of returning all the rows from the EMPLOYEES table since the EMPLOYEE\_ID column contains unique values for each employee record. Additionally, **C** returns two columns, which is not what was originally requested.

1. Choose the correct syntax to return all columns and rows of data from the EMPLOYEES table.

A. select all from employees;

B. select employee\_id, first\_name, last\_name, first\_name, department\_id

from employees;

C. select % from employees;

D. select \* from employees;

E. select \*.\* from employees;

✓ **D.** An asterisk is the SQL operator that implies that all columns must be selected from a

table.

˚**A, B, C,** and **E** are incorrect. **A** uses the ALL reserved word but is missing any column

specification and will, therefore, generate an error. **B** selects some columns but not all columns

and, therefore, does not answer the question. **C** and **E** make use of illegal selection operators.

1. **There are four rows of data in the REGIONS table. Consider the following SQL statement:**

SELECT '6 \* 6' “Area” FROM REGIONS;

**How many rows of results are returned and what value is returned by the Area column?** (Choose the best answer.)

A. 1 row returned, Area column contains value 36

B. 4 rows returned, Area column contains value 36 for all 4 rows

C. 1 row returned, Area column contains value 6 \* 6

D. 4 rows returned, Area column contains value 6 \* 6 for all 4 rows

E. A syntax error is returned.

**✓ D.** The literal expression '6 \* 6' is selected once for each row of data in the REGIONS table.

˚**A, B, C,** and **E** are incorrect. **A** returns one row instead of four and calculates the product6 \* 6. The enclosing quote operators render 6 \* 6 a character literal and not a numeric literal that can be calculated. **B** correctly returns four rows but incorrectly evaluates the character literal as a numeric literal. **C** incorrectly returns one row instead of four and **E** is incorrect, because the given SQL statement can be executed.

1. **Which three of the following conditions are equivalent to each other?**

A. WHERE SALARY <=5000 AND SALARY >=2000

B. WHERE SALARY IN (2000,3000,4000,5000)

C. WHERE SALARY BETWEEN 2000 AND 5000

D. WHERE SALARY > 1999 AND SALARY < 5001

E. WHERE SALARY >=2000 AND <=5000

✓ **A, C,** and **D.**

Each of these conditions tests for SALARY values in the range of $2000 to

$5000.

˚**B** and **E** are incorrect. **B** excludes values like $2500 from its set, and **E** is illegal since it is missing the SALARY column name reference after the AND operator.

1. **Which two of the following conditions are equivalent to each other?**

A. WHERE COMMISSION\_PCT IS NULL

B. WHERE COMMISSION\_PCT = NULL

C. WHERE COMMISSION\_PCT IN (NULL)

D. WHERE NOT(COMMISSION\_PCT IS NOT NULL)

✓ **A** and **D.**

The IS NULL operator correctly evaluates the COMMISSION\_PCT column for NULL values. **D** uses the NOT operator to negate the already negative version of the IS NULL operator, IS NOT NULL. Two negatives return a positive, and therefore **A** and **D** are equivalent. ˚**B** and **C** are incorrect since NULL values cannot be tested by the equality operator or the IN operator.

1. **What value is returned after executing the following statement? Take note that 01-JAN-2009 occurs on a Thursday. (Choose the best answer.)**

SELECT NEXT\_DAY('01-JAN-2009','wed') FROM DUAL;

* 1. 07-JAN-2009
  2. 31-JAN-2009
  3. Wednesday
  4. None of the above

**✓ A.** Since the first of January 2009 falls on a Thursday, the date of the following Wednesday is six days later.

˚**B, C,** and **D** are incorrect. **B** returns the last day of the month in which the given date falls, and **C** returns a character string instead of a date.

1. **What value is returned after executing the following statement?**

SELECT SUM(SALARY) FROM EMPLOYEES;

**Assume there are 10 employee records and each contains a SALARY value of 100, except for 1, which has a null value in the SALARY field. (Choose the best answer.)**

A. 900

B. 1000

C. NULL

D. None of the above

✓ **A.** The SUM aggregate function ignores null values and adds nonnull values. Since nine

rows contain the SALARY value 100, 900 is returned.

˚**B, C,** and **D** are incorrect. **B** would be returned if SUM(NVL(SALARY,100)) was

executed. **C** is a tempting choice since regular arithmetic with NULL values returns a NULL

result. However, the aggregate functions, except for COUNT(\*), ignore NULL values.

1. **Choose one correct statement regarding the following query:**

**SELECT \* FROM EMPLOYEES E**

**JOIN DEPARTMENTS D ON (D.DEPARTMENT\_ID=E.DEPARTMENT\_ID) JOIN LOCATIONS L ON (L.LOCATION\_ID =D.LOCATION\_ID);**

A. Joining three tables is not permitted.

B. A Cartesian product is generated.

C. The JOIN…ON clause may be used for joins between multiple tables.

D. None of the above

✓ **C.** The JOIN…ON clause and the other join clauses may all be used for joins between multiple tables. The JOIN…ON and JOIN…USING clauses are better suited for N-way table joins.

˚**A, B,** and **D** are incorrect. **A** is false since you may join as many tables as you wish. A Cartesian product is not created since there are two join conditions and three tables.

1. **Consider this statement:**

insert into regions (region\_id,region\_name)

values ((select max(region\_id)+1 from regions), 'Great Britain');

**What will the result be? (Choose the best answer.)**

A. The statement will not succeed if the value generated for REGION\_ID is not unique, because REGION\_ID is the primary key of the REGIONS table.

B. The statement has a syntax error because you cannot use the VALUES keyword with a subquery.

C. The statement will execute without error.

D. The statement will fail if the REGIONS table has a third column.

✓ **C.** The statement is syntactically correct, and the use of “MAX(REGION\_ID) + 1” guarantees generating a unique number for the primary key column.

˚**A, B, D. A** is wrong because the function will generate a unique value for the primary key. **B** is wrong because there is no problem using a scalar subquery to generate a value for a VALUES list. What cannot be done is to use the VALUES keyword and then a single nonscalar

subquery to provide all the values. **D** is wrong because if there is a third column, it will be populated with a NULL value.

1. **Consider this statement:**

create table t1 as select \* from regions where 1=2;

**What will be the result? (Choose the best answer.)**

A. There will be an error because of the impossible condition.

B. No table will be created because the condition returns FALSE.

C. The table T1 will be created but no rows inserted because the condition returns FALSE.

D. The table T1 will be created and every row in REGIONS inserted because the condition returns a NULL as a row filter.

✓ **C.** The condition applies only to the rows selected for insert, not to the table creation.

˚**A, B, D. A** is wrong because the statement is syntactically correct. **B** is wrong because the

condition does not apply to the DDL, only to the DML. **D** is wrong because the condition will

exclude all rows from selection.