**Joins**

1. How many rows would a two-table join produce if one table had 50,000 rows and the other had 100,000?

**-5 billion rows**

1. What type of join appears in the following select statement?

select e.name, e.employee\_id, ep.salary from employee\_tbl e, employee\_pay\_tbl ep where e.employee\_id = ep.employee\_id;

**-equi-join since you are matching all the “employee\_id”s in the two tables**

1. Will the following SELECT statements work?

select name, employee\_id, salary from employee\_tbl e, employee\_pay\_tbl ep where employee\_id = employee\_id and name like '%MITH';

select e.name, e.employee\_id, ep.salary from employee\_tbl e, employee\_pay\_tbl ep where name like '%MITH';

select e.name, e.employee\_id, ep.salary from employee\_tbl e, employee\_pay\_tbl ep where e.employee\_id = ep.employee\_id and e.name like '%MITH';

**-the third one. Reason: 1st one -> columns and tables aren’t named correctly. 2nd one -> “join” command missing**

1. In the WHERE clause, when joining the tables, should you do the join first or the conditions?

**-joins go before the conditions**

1. In joining tables are you limited to one-column joins, or can you join on more than one column?

**-You can join one more column**

1. Rewrite the following query to make it more readable and shorter.

select orders.orderedon, orders.name, part.partnum, part.price, part.description from orders, part where orders.partnum = part.partnum and orders.orderedon between '1-SEP-96' and '30-SEP-96' order by part.partnum;

**select o.orderedon ORDER\_DATE, o.name NAME, p.partnum**

**PART#, p.price PRICE, p.description DESCRIPTION**

**from orders o, part p**

**where o.partnum = p.partnum**

**and o.partnum like ‘%SEP%’**

**order by ORDER\_DATE;**

**SUBQUERIES: The Embedded SELECT Statement**

Are the following statements true or false?

1. The aggregate functions SUM, COUNT, MIN, MAX, and AVG all return multiple values.

**-false**

1. The maximum number of subqueries that can be nested is two.

**-false**

1. Correlated subqueries are completely self-contained.

**-false**

1. Will the following subqueries work using the ORDERS table and the PART table?

SELECT \* FROM PART;

| **PARTNUM** | **DESCRIPTION** | **PRICE** |
| --- | --- | --- |
| 54 | PEDALS | 54.25 |
| 42 | SEATS | 24.50 |
| 46 | TIRES | 15.25 |
| 23 | MOUNTAIN BIKE | 350.45 |
| 76 | ROAD BIKE | 530.00 |
| 10 | TANDEM | 1200.00 |

SELECT \* FROM ORDERS;

| **ORDEREDON** | **NAME** | **PARTNUM** | **QUANITY** | **REMARKS** |
| --- | --- | --- | --- | --- |
| 15-MAY-96 | TRUE WHEEL | 23 | 6 | PAID |
| 19-MAY-96 | TRUE WHEEL | 76 | 3 | PAID |
| 2-SEP-96 | TRUE WHEEL | 10 | 1 | PAID |
| 30-JUN-96 | BIKE SPEC | 54 | 10 | PAID |
| 30-MAY-96 | BIKE SPEC | 10 | 2 | PAID |
| 30-MAY-96 | BIKE SPEC | 23 | 8 | PAID |
| 17-JAN-96 | BIKE SPEC | 76 | 11 | PAID |
| 17-JAN-96 | LE SHOPPE | 76 | 5 | PAID |
| 1-JUN-96 | LE SHOPPE | 10 | 3 | PAID |
| 1-JUN-96 | AAA BIKE | 10 | 1 | PAID |
| 1-JUN-96 | AAA BIKE | 76 | 4 | PAID |
| 1-JUN-96 | AAA BIKE | 46 | 14 | PAID |
| 11-JUL-96 | JACKS BIKE | 76 | 14 | PAID |

* A) SELECT \* FROM ORDERS WHERE PARTNUM = SELECT PARTNUM FROM PART WHERE DESCRIPTION = 'TRUE WHEEL';

**-No, its missing parenthesis around the subquery**

* B) SELECT PARTNUM FROM ORDERS WHERE PARTNUM = (SELECT \* FROM PART WHERE DESCRIPTION = 'LE SHOPPE');

**-No**

* C) SELECT NAME, PARTNUM FROM ORDERS WHERE EXISTS (SELECT \* FROM ORDERS WHERE NAME = 'TRUE WHEEL');

**-Yes**

**Manipulating Data**

1. What is wrong with the following statement?

DELETE COLLECTION;

**-Its missing “from” in the middle in order to delete all records from that table.**

1. What is wrong with the following statement?

INSERT INTO COLLECTION SELECT \* FROM TABLE\_2

**-The “into” and “insert” statement is the problem. You would need to separate into two statements.**

**INSERT COLLECTION**

**SELECT \* FROM TABLE\_2**

1. What is wrong with the following statement?

UPDATE COLLECTION ("HONUS WAGNER CARD", 25000, "FOUND IT");

**- There is a conflict with the “update” and “insert” function.**

**UPDATE COLLECTIONS**

**SET NAME – “HONUS WAGNER CARD”,**

**VALUE = 25000,**

**REMARKS = “FOUND IT”;**

1. What would happen if you issued the following statement?

DELETE \* FROM COLLECTION;

**-Nothing will happen.**

1. What would happen if you issued the following statement?

UPDATE COLLECTION SET WORTH = 555 SET REMARKS = 'UP FROM 525';

**-All values in the worth column are now 555. All remarks now say “UP FROM 525”**

1. Will the following SQL statement work?

INSERT INTO COLLECTION SET VALUES = 900 WHERE ITEM = 'STRING';

**-There is conflict with “INSERT” and “SET” functions. It will not work.**

1. Will the following SQL statement work?

UPDATE COLLECTION SET VALUES = 900 WHERE ITEM = 'STRING';

-Yes

1. Try inserting values with incorrect data types into a table. Note the errors and then insert values with correct data types into the same table.

-The errors are caused by the incompatible data type.

1. Using your database system, try exporting a table (or an entire database) to some other format. Then import the data back into your database. Familiarize yourself with this capability. Also, export the tables to another database format if your DBMS supports this feature. Then use the other system to open these files and examine them.

**Creating and Maintaining Tables**

1. **True or False:** The ALTER DATABASE statement is often used to modify an existing table's structure.

**-False, some databases do not have that command**

1. **True or False:** The DROP TABLE command is functionally equivalent to the DELETE FROM <table name> command.

**-False. The DROP TABLE command deletes the table from the database. DELETE FROM command deletes the records from the table.**

|  |
| --- |
|  |

1. **True or False:** To add a new table to a database, use the CREATE TABLE command.

**-True**

1. What is wrong with the following statement?

CREATE TABLE new\_table ( ID NUMBER, FIELD1 char(40), FIELD2 char(80), ID char(40);

**- Closing parenthesis are missing at the end of the statement. ID keeps getting repeated.**

1. What is wrong with the following statement?

ALTER DATABASE BILLS ( COMPANY char(80));

**-You need to used ALTER TABLE command**

1. When a table is created, who is the owner?

-Whoever created the table is the owner.

1. If data in a character column has varying lengths, what is the best choice for the data type?

-VARCHAR2

1. Add two tables to the BILLS database named BANK and ACCOUNTTYPE using any format you like. The BANK table should contain information about the BANK field used in the BANKACCOUNTStable in the examples. The ACCOUNTTYPE table should contain information about the ACCOUNTTYPE field in the BANKACCOUNTS table also. Try to reduce the data as much as possible.

You should use the CREATE TABLE command to make the tables. Possible SQL statements would look like this:

CREATE TABLE BANK ( ACCOUNT\_ID NUMBER(30) NOT NULL, BANK\_NAME VARCHAR2(30) NOT NULL, ST\_ADDRESS VARCHAR2(30) NOT NULL, CITY VARCHAR2(15) NOT NULL, STATE CHAR(2) NOT NULL, ZIP NUMBER(5) NOT NULL; CREATE TABLE ACCOUNT\_TYPE ( ACCOUNT\_ID NUMBER(30) NOT NULL, SAVINGS CHAR(30), CHECKING CHAR(30);

CREATE TABLE ACCOUNT\_TYPE

(ACCOUNT\_ID NUMBER(30) NOT NULL,

SAVINGS CHAR(30),

CHECKING CHAR(30);