[SQL, ASSIGNMENT]

[This Assignment has been designed keeping the level of training sessions in mind . Anybody who wants to practice about queries and creating tables in MS SQL should go through this.]

SQL Assignment

Directions for questions 1-6. Each of the questions just gives a list of SQL data types. In some cases the list may consist of what the book calls synonyms. In other cases the types may not be synonyms, although they are related. For each question give a brief statement of what kind of data can be stored in the list of types given.

1. CHAR(n), TEXT(n), VARCHAR(n).

Alphanumeric data either fixed at n symbols or up to n symbols. It's not possible to do arithmetic on this data.

2. REAL, FLOAT, NUMBER, NUMERIC, DECIMAL.

These are numbers with decimal places.

3. INTEGER, LONG, INT, SMALLINT.

These are all whole numbers. They vary in the how big a number they can hold.

4. MONEY, CURRENCY.

These are numeric types with decimal places for holding monetary values.

5. BINARY, LONGBINARY, GENERAL, IMAGE, OLEOBJECT.

These can hold complete files, such as pictures or media. They are not of a fixed in size, and the upper limit on their size is large.

6. DATE, TIME, DATETIME.

These can hold date and time values. These are hybrid types. Although you think of them as text type fields, it is possible to do arithmetic and numeric comparisons on them.

7. Write an SQL command that will create a table named Friend with the following fields and types: idno CHAR(3), name CHAR(24), address CHAR(24), birthday DATE, waistsize INTEGER, giftvalue CURRENCY.

```
CREATE TABLE Friend
(idno CHAR(3),
name CHAR(24),
address CHAR(24),
birthday DATE
waistsize INTEGER,
giftvalue CURRENCY)
```

8. Write an SQL command that will create a table named Friend with the following fields: idno, name, address, age, bankbalance. Each of these fields has characteristics that would make a particular type appropriate for it. You need to choose the type and size for the fields.

```
CREATE TABLE Friend
(idno CHAR(3),
name CHAR(24),
address CHAR(24),
age INTEGER,
bankbalance CURRENCY)
```

9. Write an SQL command that will put this name into a record in the Friend table: 'Road Runner'.

```
INSERT INTO Friend(name)
VALUES ('Road Runner')
```

10. For the record created by the previous question, what values would be in the fields other than the name field?

NULL

11. Write an SQL command that will insert a complete record into the Friend table with these values for the respective fields: '123', 'Tasmanian Devil', 'Tasmania', #07/07/57#, 32, 29.99.

```
INSERT INTO Friend
VALUES('123', 'Tasmanian Devil', 'Tasmania', #07/07/57#, 32, 29.99)
```

12. Write an SQL command that will insert a complete record into the Friend table with these values for the respective fields: '456', 'Felix the Cat', 'Hollywood', NULL, NULL, NULL.

```
INSERT INTO Friend
VALUES('456', 'Felix the Cat', 'Hollywood', NULL, NULL, NULL)
```

13. Write an SQL command (with a query included) that will insert into Friend all of the spno's, names, and addresses from the salesperson table.

```
INSERT INTO Friend(idno, name, address)
SELECT spno, name, addr
FROM Salesperson
```

14. Write an SQL command (with a query included) that will insert into Friend all of the custno's, names, and addresses of customers where the state is equal to 'WA'.

```
INSERT INTO Friend(idno, name, address)
SELECT custno, name, addr
FROM Customer
WHERE state = 'WA'
```

15. Write an SQL command (with a query included) that will insert into Friend just the custno's of all of the customers where the state is equal to 'WA'.

```
INSERT INTO Friend(idno)
SELECT custno
FROM Customer
WHERE state = 'WA'
```

16. Write an SQL command (with a query included) that will insert into Friend just the spno's and names of salespeople.

```
INSERT INTO Friend(idno, name)
SELECT spno, name
FROM Salesperson
```

17. What's wrong with this query?

```
INSERT INTO Friend(idno, name, giftvalue)
SELECT spno, name, commrate
FROM salesperson
```

It doesn't make sense to put commrate into giftvalue.

18. What’s wrong with this query?

```
INSERT INTO Friend
SELECT *
FROM Salesperson
```

There aren’t as many fields in the Friend table as there are in the Salesperson table, so it’s not possible to select all from Salesperson to insert into Friend. Plus, the respective fields don't match anyway.

19. Write an SQL command that will update the giftvalue to 49.99 for all people in the Friend table who are 21 and older.

```
UPDATE Friend
SET giftvalue = 49.99
WHERE age >= 21
```

20. Write an SQL command that will update the Friend table so that all waistsizes will be 0.

```
UPDATE Friend
SET waistsize = 0
```

21. Write an SQL command that will add a city field to the Friend table. You can choose its type and size.

```
ALTER TABLE Friend
ADD city CHAR(12)
```

22. Write an SQL command that will change the specifications of the name field in the Friend table to CHAR(36).

```
ALTER TABLE Friend
ALTER COLUMN name CHAR(36)
```

23. Write an SQL command that will get rid of the city field from the Friend table.

```
ALTER TABLE Friend
DROP city
```

24. Write an SQL command that will delete all records from the Friend table where the giftvalue is over 100.

```
DELETE *
FROM Friend
WHERE giftvalue > 100
```

25. Write an SQL command that will drop the table Friend from a database, eliminating it and all of its records.

```
DROP TABLE Friend
```

The remaining questions refer to the salesdate field in the Carsale table.

26. A) What distinctive punctuation marks should be used to enclose date values in queries? B) What different kinds of punctuation marks can be used inside date values in order to separate month, day, and year?

A) number signs. B) Dashes and slashes are the correct separators for dates when you want to enter values into a table or give a specific value for comparison in a WHERE clause, for example. It is possible to print out dates with various punctuation marks when doing formatting.

27. Write a query that will select all fields from the Carsale table where the salesdate is after January 1st, 2006.

```
SELECT *
FROM Carsale
WHERE salesdate > #01/01/06#
```

28. Write the same query as for question 27, but use a different punctuation mark to separate the parts of the date value.

```
SELECT *
FROM Carsale
WHERE salesdate > #01-01-06#
```

29. Write an SQL query that will select the salesdate field from the Carsale table formatted as follows: month, day, and year in that order, numeric, separated by slashes. You may assume that your computer is using U.S. default settings.

This answer is not supposed to be a trick. It merely points out the fact that the format specified is the default for U.S. dates, so no separate FORMAT is needed in order to achieve the desired results.

```
SELECT salesdate
FROM Carsale
```

30. Write an SQL query that will select the salesdate field from the Carsale table formatted as follows: the full name of the day, a comma, the full name of the month, the number of the day in the month, a comma, and a 4 digit year.

```
SELECT FORMAT(salesdate, 'dddd, mmmm dd, yyyy')
FROM Carsale
```

31. Write an SQL query that will select the salesdate field from the Carsale table formatted as follows: 2 digit number of day in the month, 3 letter abbreviation of month, 2 digit year. Spaces should be used as separators.

```
SELECT FORMAT(salesdate, 'dd mmm yy')
FROM Carsale
```

32. Write an SQL query that will select the salesdate field from the Carsale table formatted as follows: 4 digit year, comma, 2 digit week of year, comma, 1 digit number of day of the week.

```
SELECT FORMAT(date, 'yyyy, ww, w')
FROM Carsale
```