**Grantham University**

**IS259 Database Applications**

**November 20, 2016**

**Does all standard SQL work in Microsoft Access? Explain.**

* SQL (pronounced "ess-que-el") stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems.
* Nearly all SQL query work with Microsoft access. We need to modify some statements to execute them as they as executed in SQL.
* When you want to retrieve data from a database, you ask for the data by using Structured Query Language, or SQL. SQL is a computer language that closely resembles English that database programs understand. Knowing SQL is important because every query in Microsoft Access uses SQL.
* Understanding how SQL works can help create better queries, and can make it easier for you to fix a query when it is not returning the results that you want.

**List and describe the four basic SQL data types**

1. **Char (n)**  
   Description: Fixed width character string. Maximum 8,000 characters  
   Storage: Defined width.
2. **Varchar (n)**  
   Description: Variable width character string. Maximum 8,000 characters  
   Storage: 2 bytes + number of chars
3. **Int**  
   Description: Allows whole numbers between -2,147,483,648 and 2,147,483,647  
   Storage: 4 bytes
4. **Bigint**  
   Description: Allows whole numbers between -9,223,372,036,854,775,808 and  
   9,223,372,036,854,775,807  
   Storage: 8 bytes

**List and describe five SQL built-in functions**

**SQL Aggregate Functions**

SQL aggregate functions return a single value, calculated from values in a column

* AVG() - Returns the average value
* COUNT() - Returns the number of rows
* FIRST() - Returns the first value
* LAST() - Returns the last value
* MAX() - Returns the largest value
* MIN() - Returns the smallest value
* SUM() - Returns the sum

## SQL Scalar functions

## SQL scalar functions return a single value, based on the input value.

* UCASE() - Converts a field to upper case
* LCASE() - Converts a field to lower case
* MID() - Extract characters from a text field
* LEN() - Returns the length of a text field
* ROUND() - Rounds a numeric field to the number of decimals specified
* NOW() - Returns the current system date and time
* FORMAT() - Formats how a field is to be display

**Write an SQL CREATE TABLE statement to create the PET\_OWNER table, with Owner ID as a surrogate key. Save as Create Pet Owner.**

CreatePetOwner:  
SQL: CREATE TABLE PET\_OWNER( OwnerID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

OwnerLastName varchar (50),

OwnerFirstName varchar (50),

OwnerPhone varchar (15),

OwnerEmail varchar (255));

**Write a set of SQL INSERT statements to populate the PET\_OWNER table with the data given above. Save as PopulatePetOwner.**

 PopulatePetOwner  
SQL: Insert INTO PET\_OWNER (OwnerID, OwnerLastName, OwnerFirstName, OwnerPhone,  
OwnerEmail)

VALUES (1,’Downs’,’Marsha’,’555-537-8765’,’Marshadowns@somewhere.com’);

Or  
Insert INTO PET\_OWNER (OwnerLastName, OwnerFirstName, OwnerPhone, OwnerEmail)  
VALUES(’Downs’,’Marsha’,’555-537-8765’,’Marshadowns@somewhere.com’ );

Note: Id will be added automatically.

**Write an SQL CREATE TABLE statement to create the PET table, with PetID as a surrogate key. Save as CreatePet.**

CreatePet:  
SQL: CREATE TABLE PET(

PetID INT NOT NULL AUTO\_INCREMENT\_PRIMARY KEY,

PetName varchar (50),

PetType varchar (50),

PetBreed varchar (30),

PetDOB varchar (15),

OwnerId int NOT NULL,  
FOREIGN KEY (OwnerID) REFERENCES PET\_OWNER(OwnerID));

**Write a set of SQL INSERT statements to populate the PET table with the data given above. Save as PopulatePet**

NSERT INTO PET(PetID, PetName, PetType, PetBreed, PetDOB, OwnerId)  
VALUES(1,’King’,’Dog’,’std.Poddle’,’27-feb-2011’,1);

**Write an SQL statement to display the breed and type of all pets. Save as AllBreeds**

 AllBreeds  
SQL: Select PetType, PetBreed form PET;

**Write an SQL statement to display the breed, and DOB of all pets having the type Cat. Save as Cats**

Cats  
SQL: Select PetBreed, PetDOB from PET where PetType=’Cat’;

**Write an SQL statement to display the first name, last name, and email of all owners, sorted in alphabetical order by last name. Save as AlphaOwners**

AlphaOwners  
SQL: select OwnerFirstName, OwnerLastName, OwnerEmail from pet\_owner order by  
OwnerLastName ASC;

**Write an SQL statement to display all the owners’ names, with the first name in all lower case and the last name in all upper case. Save as UpperLower**

UpperLower  
SQL: select LCASE (OwnerFirstName), UCASE (OwnerLastName) from pet\_owner;

**Write an SQL statement to display the total number of pets. Save as TotalPets**

TotalPets  
SQL: SELECT COUNT (\*) FROM PET;

**Write an SQL statement to display the last name, first name and email of any owner who has a NULL value for OwnerPhone. (Note: there should be one owner who has a NULL value for OwnerPhone.) Save as PhoneNull**

PhoneNull  
SQL: SELECT OwnerFirstName, OwnerLastName, OwnerEmail from PET\_OWNER where  
OwnerPhone= '‘;

**Write an SQL statement to count the number of distinct breeds. Save as NumberOfBreeds**

NumberOfBreeds  
SQL: Select Count (Distinct PetBreed) from PET;

**Write an SQL statement to display the names of all the dogs. Save as Dogs**

Dogs  
SQL: Select PetName from PET where PetType=’Dog’ ;