Purpose: to give students a chance to use SQL DDL and update commands.

Requirements: do all of the following in your personal database schema; i.e., s7Yournetid (case matters) and provide evidence below on what you did and how you did it. I will also review your database so don’t make any other changes to it before I grade the assignment.

Make your database the default database (show SQL).

**Data Definition**

1. Create a Category table exactly like the one shown below.



**Create the table**

SQL DDL:

CREATE TABLE `s7MichaeliRM30`.`Category` (

`categoryId` INT(11) NOT NULL,

`categoryName` VARCHAR(45) NULL,

`description` VARCHAR(100) NULL,

PRIMARY KEY (`categoryId`));

**Show that table was created**

SQL:

show tables;

Results:

# Tables\_in\_s7MichaeliRM30

Category

**Show the table definition**

SQL:

describe Category;

Results:

# Field, Type, Null, Key, Default, Extra

categoryId, int(11), NO, PRI, ,

categoryName, varchar(45), YES, , ,

description, varchar(100), YES, , ,

2. Create a Supplier table exactly like the one shown below.



**Create the table**

SQL DDL:

CREATE TABLE `s7MichaeliRM30`.`Supplier` (

`supplierId` INT(11) NOT NULL,

`companyName` VARCHAR(45) NOT NULL,

`address` VARCHAR(60) NULL,

`city` VARCHAR(45) NULL,

`region` VARCHAR(20) NULL,

`postalCode` VARCHAR(10) NULL,

`country` VARCHAR(45) NULL,

`homePage` VARCHAR(60) NULL,

`contactFirstName` VARCHAR(45) NOT NULL,

`contactLastName` VARCHAR(45) NOT NULL,

`contactTitle` VARCHAR(45) NULL,

`phone` VARCHAR(18) NOT NULL,

`fax` VARCHAR(18) NULL,

PRIMARY KEY (`supplierId`));

**Show that table was created**

SQL:

show tables;

Results:

# Tables\_in\_s7MichaeliRM30

Category

Supplier

**Show the table definition**

SQL:

describe Supplier;

Results:

# Field, Type, Null, Key, Default, Extra

supplierId, int(11), NO, PRI, ,

companyName, varchar(45), NO, , ,

address, varchar(60), YES, , ,

city, varchar(45), YES, , ,

region, varchar(20), YES, , ,

postalCode, varchar(10), YES, , ,

country, varchar(45), YES, , ,

homePage, varchar(60), YES, , ,

contactFirstName, varchar(45), NO, , ,

contactLastName, varchar(45), NO, , ,

contactTitle, varchar(45), YES, , ,

phone, varchar(18), NO, , ,

fax, varchar(18), YES, , ,

3. Create a Product table exactly like the one shown below. Make sure you do the foreign keys. Set them up so there is no action if the parent rows are deleted.



**Create the table**

SQL DDL:

CREATE TABLE `s7MichaeliRM30`.`Product` (

`productId` INT(11) NOT NULL,

`prodName` VARCHAR(45) NOT NULL,

`stdUnitPrice` DECIMAL(7,2) NULL,

`supplierId` INT(11) NULL,

`categoryId` INT(11) NULL,

PRIMARY KEY (`productId`),

INDEX `supplierId\_idx` (`supplierId` ASC),

INDEX `categoryId\_idx` (`categoryId` ASC),

CONSTRAINT `supplierId`

FOREIGN KEY (`supplierId`)

REFERENCES `s7MichaeliRM30`.`Supplier` (`supplierId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `categoryId`

FOREIGN KEY (`categoryId`)

REFERENCES `s7MichaeliRM30`.`Category` (`categoryId`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

**Show that table was created**

SQL:

show tables;

Results:

# Tables\_in\_s7MichaeliRM30

Category

Product

Supplier

**Show the table definition**

SQL:

describe Product;

Results:

# Field, Type, Null, Key, Default, Extra

productId, int(11), NO, PRI, ,

prodName, varchar(45), NO, , ,

stdUnitPrice, decimal(7,2), YES, , ,

supplierId, int(11), YES, MUL, ,

categoryId, int(11), YES, MUL, ,

4. Modify the Product table to match what is shown below.



**Modify the table**

SQL DDL:

ALTER TABLE `s7MichaeliRM30`.`Product`

ADD COLUMN `qtyPerUnit` VARCHAR(25) NULL AFTER `stdUnitPrice`,

ADD COLUMN `unitsInStock` INT(11) NULL AFTER `qtyPerUnit`,

ADD COLUMN `unitsInOrder` INT(11) NULL AFTER `unitsInStock`,

ADD COLUMN `reorderLevel` INT(11) NULL AFTER `unitsInOrder`,

ADD COLUMN `status` CHAR(1) NOT NULL AFTER `reorderLevel`;

**Show the new table definition**

SQL:

describe Product;

Results:

# Field, Type, Null, Key, Default, Extra

productId, int(11), NO, PRI, ,

prodName, varchar(45), NO, , ,

stdUnitPrice, decimal(7,2), YES, , ,

qtyPerUnit, varchar(25), YES, , ,

unitsInStock, int(11), YES, , ,

unitsInOrder, int(11), YES, , ,

reorderLevel, int(11), YES, , ,

status, char(1), NO, , ,

supplierId, int(11), YES, MUL, ,

categoryId, int(11), YES, MUL, ,

5. Add indexes to the Supplier table for city and contactLastName.

**Create the indexs**

SQL DDL:

ALTER TABLE `s7MichaeliRM30`.`Supplier`

ADD INDEX `city` (`city` ASC),

ADD INDEX `contactLastName` (`contactLastName` ASC);

**Show the index**

SQL:

show indexes from Supplier;

Results:

# Table, Non\_unique, Key\_name, Seq\_in\_index, Column\_name, Collation, Cardinality, Sub\_part, Packed, Null, Index\_type, Comment, Index\_comment

Supplier, 0, PRIMARY, 1, supplierId, A, 0, , , , BTREE, ,

Supplier, 1, city, 1, city, A, 0, , , YES, BTREE, ,

Supplier, 1, contactLastName, 1, contactLastName, A, 0, , , , BTREE, ,

6. Insert 3 rows into Supplier table. Make yourself the contact person for the first supplier. Include values for all fields.

SQL: (to enter values – 1 query)

INSERT INTO Supplier

VALUES

(0001, 'Warhawk Supply', '800 W. Starin Rd.', 'Whitewater', 'WI', '53190', 'USA', 'WHS.com', 'Ryan', 'Michaelis', 'Sales Representative', '1111111111', '1111111112'),

(0002, 'Whitewater Resource', '1008 W. Starin Rd.', 'Whitewater', 'WI', '53190', 'USA', 'WWR.com', 'Tom', 'Gantry', 'Representative', '1111111113', '1111111114'),

(0003, 'Southern Wisconsin Emporium', '1020 W. Starin Rd.', 'Whitewater', 'WI', '53190', 'USA', 'SWE.com', 'Rick', 'Mando', 'Sales Manager', '1111111115', '1111111116');

SQL: (to prove data was entered – 1 query)

SELECT \* FROM Supplier;

Results:

# supplierId, companyName, address, city, region, postalCode, country, homePage, contactFirstName, contactLastName, contactTitle, phone, fax

1, Warhawk Supply, 800 W. Starin Rd., Whitewater, WI, 53190, USA, WHS.com, Ryan, Michaelis, Sales Representative, 1111111111, 1111111112

2, Whitewater Resource, 1008 W. Starin Rd., Whitewater, WI, 53190, USA, WWR.com, Tom, Gantry, Representative, 1111111113, 1111111114

3, Southern Wisconsin Emporium, 1020 W. Starin Rd., Whitewater, WI, 53190, USA, SWE.com, Rick, Mando, Sales Manager, 1111111115, 1111111116

7. Insert 2 rows into Category table. Include values for all fields.

SQL: (to enter values – 1 query)

INSERT INTO Category

VALUES

(010, 'Construction', 'Supplies for actual structure.'),

(011, 'Electrical', 'Supplies for electrical work.');

SQL: (to prove data was entered – 1 query)

Select \* FROM Category;

Results:

# categoryId, categoryName, description

10, Construction, Supplies for actual structure.

11, Electrical, Supplies for electrical work.

8. Insert 4 rows into Product table. Include values for all fields. (Don’t forget about referential integrity).

SQL: (to enter values – 1 query)

INSERT INTO Product

VALUES

(1, 'Copper Wire 10ft', 24.98, '2', 213, 14, 40, 'A', 1, 11),

(2, 'Dimmer Switches', 48.98, '3', 84, 34, 21, 'A', 2, 11),

(3, 'Pine 2x4x10', 8.00, '1', 234, 162, 80, 'U', 3, 10),

(4, 'Drywall 8x5x1"', 80.22, '2', 38, 12, 12, 'A', 1, 10);

SQL: (to prove data was entered – 1 query)

SELECT \* FROM Product;

Results:

# productId, prodName, stdUnitPrice, qtyPerUnit, unitsInStock, unitsInOrder, reorderLevel, status, supplierId, categoryId

1, Copper Wire 10ft, 24.98, 2, 213, 14, 40, A, 1, 11

2, Dimmer Switches, 48.98, 3, 84, 34, 21, A, 2, 11

3, Pine 2x4x10, 8.00, 1, 234, 162, 80, U, 3, 10

4, Drywall 8x5x1", 80.22, 2, 38, 12, 12, A, 1, 10

9. Give yourself a promotion to Director of Sales.

SELECT \*

FROM Supplier

WHERE supplierId = 1;

SQL: (to change values – more than one query)

UPDATE Supplier

SET contactTitle = 'Director of Sales'

WHERE supplierId = 1;

SQL: (to show that changes have taken place – use one query)

SELECT \*

FROM Supplier

WHERE supplierId = 1;

Results:

# supplierId, companyName, address, city, region, postalCode, country, homePage, contactFirstName, contactLastName, contactTitle, phone, fax

1, Warhawk Supply, 800 W. Starin Rd., Whitewater, WI, 53190, USA, WHS.com, Ryan, Michaelis, Director of Sales, 1111111111, 1111111112

10. Delete one of the products.

SELECT \*

FROM Product

WHERE productId = 2;

SQL: (to do the delete)

DELETE

FROM Product

WHERE productId = 2;

SQL: (to show that the delete took place)

SELECT \* FROM Product;

Results:

# productId, prodName, stdUnitPrice, qtyPerUnit, unitsInStock, unitsInOrder, reorderLevel, status, supplierId, categoryId

1, Copper Wire 10ft, 24.98, 2, 213, 14, 40, A, 1, 11

3, Pine 2x4x10, 8.00, 1, 234, 162, 80, U, 3, 10

4, Drywall 8x5x1", 80.22, 2, 38, 12, 12, A, 1, 10

**Do not make any other changes to your database until I have had a chance to grade the assignments.**