

---

CSC 675/755-02 Database Project

Phase 2 Final Report

**Spring 2021**

---

Cindy Fung Chan

Rhetta Mae Sevilla

Utkrisht Sharma

Abraham Zepeda

## Create Database:

```
CREATE DATABASE csc675;
```

## Use Database:

```
USE csc675;
```

## Create Tables, Indexes and Constraints:

### Tables

```
CREATE TABLE User (  
  name varchar(20) NOT NULL,  
  username varchar(20) NOT NULL,  
  u_id varchar(20) NOT NULL,  
  login varchar(20) NOT NULL,  
  password varchar(20) NOT NULL,  
  phone varchar(20) NOT NULL,  
  address varchar(20) NOT NULL,  
  email varchar(20) NOT NULL,  
  PRIMARY KEY (u_id)  
);
```

```
CREATE TABLE Seller (  
  u_id varchar(20) NOT NULL,  
  post varchar(20) DEFAULT NULL,  
  profile_pic blob,  
  about_seller varchar(20) DEFAULT NULL,  
  PRIMARY KEY (u_id),  
  CONSTRAINT Seller_ibfk_1 FOREIGN KEY (u_id) REFERENCES User (u_id)  
);
```

```
CREATE TABLE Buyer (  
  u_id varchar(20) NOT NULL,  
  comment varchar(20) DEFAULT NULL,  
  PRIMARY KEY (u_id),  
  CONSTRAINT Buyer_ibfk_1 FOREIGN KEY (u_id) REFERENCES User (u_id)  
);
```

```
CREATE TABLE Product (  
  p_id varchar(20) NOT NULL,
```

```

picture varchar(200) DEFAULT NULL,
p_name varchar(20) NOT NULL,
price FLOAT NOT NULL,
category varchar(20) NOT NULL,
quantity INTEGER(20) NOT NULL,
p_description varchar(20) NOT NULL,
u_id varchar(20) NOT NULL,
PRIMARY KEY (p_id),
KEY u_id (u_id),
CONSTRAINT Product_ibfk_1 FOREIGN KEY (u_id) REFERENCES User (u_id)
);

```

```

CREATE TABLE Cart (
  cart_id varchar(20) NOT NULL,
  p_name varchar(20) NOT NULL,
  quantity INTEGER(20) NOT NULL,
  price FLOAT NOT NULL,
  total_cost FLOAT NOT NULL,
  u_id varchar(20) DEFAULT NULL,
PRIMARY KEY (cart_id),
UNIQUE KEY u_id (u_id),
CONSTRAINT Cart_ibfk_1 FOREIGN KEY (u_id) REFERENCES Buyer (u_id) ON
DELETE CASCADE ON UPDATE NO ACTION
);

```

```

CREATE TABLE Ordered (
  p_name varchar(20) NOT NULL,
  quantity int(20) NOT NULL,
  price float NOT NULL,
  p_id varchar(20) NOT NULL,
  transaction_id varchar(20) NOT NULL,
  cart_id varchar(20) NOT NULL,
PRIMARY KEY (transaction_id),
KEY cart_id (cart_id),
CONSTRAINT Ordered_ibfk_1 FOREIGN KEY (cart_id) REFERENCES Cart (cart_id) ON
DELETE CASCADE ON UPDATE NO ACTION
);

```

```

CREATE TABLE Has (
  u_id varchar(20) NOT NULL,

```

```

p_id varchar(20) NOT NULL,
KEY u_id (u_id),
KEY p_id (p_id),
CONSTRAINT Has_ibfk_1 FOREIGN KEY (u_id) REFERENCES User (u_id) ON
DELETE CASCADE ON UPDATE NO ACTION,
CONSTRAINT Has_ibfk_2 FOREIGN KEY (p_id) REFERENCES Product (p_id) ON
DELETE CASCADE ON UPDATE NO ACTION
);

```

```

CREATE TABLE Search (
  u_id varchar(20) NOT NULL,
  p_id varchar(20) NOT NULL,
  UNIQUE KEY u_id (u_id, p_id),
  UNIQUE KEY u_id_2 (u_id, p_id)
);

```

```

CREATE TABLE Added (
  u_id varchar(20) NOT NULL,
  cart_id varchar(20) NOT NULL,
  p_id varchar(20) NOT NULL
);

```

```

CREATE TABLE Place (
  transaction_id varchar(20) NOT NULL,
  cart_id varchar(20) NOT NULL
);

```

### View

```

CREATE VIEW LowStockProduct
AS SELECT P.p_name, P.category
FROM Product P
WHERE P.quantity <= 5;

```

### Index

This version of CREATE INDEX gave a syntax error for some reason:

```

CREATE INDEX IndCategPrice
ON Product
WITH STRUCTURE = BTREE
KEY = (category, price);

```

This version of CREATE INDEX worked:

```
CREATE INDEX IndCategPrice
ON Product (category, price)
USING BTREE;
```

## Inserting Data into the Tables:

```
INSERT INTO User (name, username, u_id, login, password, phone, address, email) VALUES
('Gina Stevens', 'gina', '401', 'user1', 'pwbuyer1', '5102233321', '31 4th st', 'stu1@sfsu.edu'),
('Samantha Smith', 'sam', '402', 'user2', 'pwbuyer2', '9252277321', '32 5th st', 'stu2@sfsu.edu'),
('Will Miller', 'will', '403', 'user3', 'pwbuyer3', '4152233333', '33 6th st', 'stu3@sfsu.edu'),
('Don Escobar', 'descobar', '301', 'user4', 'pwseller1', '4151234567', '34 6th st', 'stu4@sfsu.edu'),
('Eduardo Barnaby', 'bbarnaby', '302', 'user5', 'pwseller2', '4269853705', '35 6th st',
'stu5@sfsu.edu'), ('Liz Cortez', 'itsLiz', '303', 'user6', 'pwseller3', '4151237891', '36 6th st',
'stu6@sfsu.edu');
```

```
INSERT INTO Seller (u_id, post, profile_pic, about_seller) VALUES
('301', 'I sell good stuff', 'images/seller1.jpg', 'cool seller'),
('302', 'Buy my products', 'images/seller2.jpg', 'best seller'),
('303', 'Enjoy buying', 'images/seller3.jpg', 'great seller');
```

```
INSERT INTO Buyer (u_id, comment) VALUES
('401', 'Good chocolate'),
('402', 'It's good'),
('403', 'best chocolate');
```

```
INSERT INTO Product (p_id, picture, p_name, price, category, quantity, p_description, u_id)
VALUES
('101', 'images/1.jpg', 'chocolate', 15, 'chocolate', 1000, 'good product', '301'),
('102', 'images/2.jpg', 'almond', 20, 'chocolate', 1000, 'smooth and good', '301'),
('103', 'images/3.jpg', 'tangy', 4, 'chocolate', 1000, 'tastes very well', '301'),
('104', 'images/4.jpg', 'black', 10, 'chocolate', 1000, 'special occasions', '302'),
('105', 'images/5.jpg', 'roasted', 5, 'chocolate', 1000, 'best choc flavor', '302'),
('106', 'images/6.jpg', 'coconut', 11, 'chocolate', 1000, 'every day goodness', '301'),
('107', 'images/7.jpg', 'rose', 100, 'party', 1000, 'large size', '302'),
('108', 'images/8.jpg', 'party', 2, 'party', 1000, 'large size', '301'),
('109', 'images/9.jpg', 'tasty', 5, 'party', 1000, 'large size', '303'),
('110', 'images/10.jpg', 'large', 30, 'party', 1000, 'large size', '303');
```

```
INSERT INTO Cart (cart_id, p_name, quantity, price, total_cost, u_id) VALUES
('1', 'chocolate', 2, 15, 30, '401'),
('2', 'almond', 10, 20, 200, '402'),
('3', 'tangy', 15, 4, 60, '403');
```

```
INSERT INTO Ordered (p_name, quantity, price, p_id, transaction_id, cart_id) VALUES  
('chocolate', 2, 15, '101', '201', '1'),  
('almond', 10, 20, '102', '202', '2'),  
('tangy', 15, 4, '103', '203', '3');
```

## SQL Queries:

### Queries involving HAVING, GROUP BY, AGGREGATE OP.:

1) Find the most expensive product from each seller

```
SELECT P.u_id, MAX(P.price)  
FROM Product P  
GROUP BY P.u_id;
```

2) Find users id and username that sell more than two products

```
SELECT U.u_id, U.username  
FROM User U  
WHERE U.u_id IN ( SELECT P.u_id  
FROM Product P  
GROUP BY P.u_id  
HAVING COUNT(*) > 2);
```

### Nested queries involving IN, EXIST, op ANY, op ALL:

3) Find users that sell products that cost \$5

```
SELECT * FROM User U  
WHERE EXISTS ( SELECT P.u_id  
    FROM Product P  
    WHERE P.u_id = U.u_id  
    AND P.price = 5);
```

4) Find the most expensive product

```
SELECT * FROM Product P  
WHERE P.price >= ALL ( SELECT P.price  
FROM Product P);
```

## Snapshots:

### Table structure of csc675 database:

```
[mysql> SHOW TABLES;
+-----+
| Tables_in_csc675 |
+-----+
| Added            |
| Buyer            |
| Cart             |
| Has              |
| lowstockproduct  |
| Ordered          |
| Place            |
| Product          |
| Search           |
| Seller           |
| User             |
+-----+
11 rows in set (0.00 sec)
```

```
[mysql> DESCRIBE ADDED;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| u_id  | varchar(20)   | NO   |     | NULL    |       |
| cart_id | varchar(20)  | NO   |     | NULL    |       |
| p_id  | varchar(20)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
[mysql> DESCRIBE BUYER;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| u_id  | varchar(20)   | NO   | PRI | NULL    |       |
| comment | varchar(20)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
[mysql> DESCRIBE CART;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| cart_id | varchar(20)  | NO   | PRI | NULL    |       |
| p_name  | varchar(20)  | NO   |     | NULL    |       |
| quantity | int          | NO   |     | NULL    |       |
| price   | float        | NO   |     | NULL    |       |
| total_cost | float      | NO   |     | NULL    |       |
| u_id    | varchar(20)  | YES  | UNI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
[mysql> DESCRIBE HAS;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| u_id  | varchar(20)   | NO   | MUL | NULL    |       |
| p_id  | varchar(20)   | NO   | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
[mysql> DESCRIBE LOWSTOCKPRODUCT;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| p_name | varchar(20)   | NO   |     | NULL    |       |
| category | varchar(20)  | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
[mysql> DESCRIBE ORDERED;
```

Field	Type	Null	Key	Default	Extra
p_name	varchar(20)	NO		NULL	
quantity	int	NO		NULL	
price	float	NO		NULL	
p_id	varchar(20)	NO		NULL	
transaction_id	varchar(20)	NO	PRI	NULL	
cart_id	varchar(20)	NO	MUL	NULL	

6 rows in set (0.01 sec)

```
[mysql> DESCRIBE PLACE;
```

Field	Type	Null	Key	Default	Extra
transaction_id	varchar(20)	NO		NULL	
cart_id	varchar(20)	NO		NULL	

2 rows in set (0.00 sec)

```
[mysql> DESCRIBE PRODUCT;
```

Field	Type	Null	Key	Default	Extra
p_id	varchar(20)	NO	PRI	NULL	
picture	varchar(200)	YES		NULL	
p_name	varchar(20)	NO		NULL	
price	float	NO		NULL	
category	varchar(20)	NO	MUL	NULL	
quantity	int	NO		NULL	
p_description	varchar(20)	NO		NULL	
u_id	varchar(20)	NO	MUL	NULL	

8 rows in set (0.00 sec)

```
[mysql> DESCRIBE SEARCH;
```

Field	Type	Null	Key	Default	Extra
u_id	varchar(20)	NO	PRI	NULL	
p_id	varchar(20)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
[mysql> DESCRIBE SELLER;
```

Field	Type	Null	Key	Default	Extra
u_id	varchar(20)	NO	PRI	NULL	
post	varchar(20)	YES		NULL	
profile_pic	blob	YES		NULL	
about_seller	varchar(20)	YES		NULL	

4 rows in set (0.00 sec)

```
[mysql> DESCRIBE USER;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	NO		NULL	
username	varchar(20)	NO		NULL	
u_id	varchar(20)	NO	PRI	NULL	
login	varchar(20)	NO		NULL	
password	varchar(20)	NO		NULL	
phone	varchar(20)	NO		NULL	
address	varchar(20)	NO		NULL	
email	varchar(20)	NO		NULL	

8 rows in set (0.00 sec)







