Lab Report Cover Page

Course Number	CPS510
Course Title	Database Systems I
Semester/Year	F2020
Instructor	Dr. A. Abhari
TA Name	Haytham Qushtom

Assignment/Lab Report No.	4
Report Title	Demo of Designing Views/Simple Queries

Section No.	8
Group No.	9
Submission Date	October 27, 2020
Due Date	October 27, 2020

Student Name	Student ID	Initial
Tanvir Billah	500829695	T.B.
James Choi	500931983	J.C.
Mahamudul Islam	500963051	M.I.

```sql
CREATE TABLE customer(
customer_id NUMBER PRIMARY KEY,
customer_name VARCHAR2(128) NOT NULL,
customer_email VARCHAR2(255) NOT NULL ,
customer_phone VARCHAR2(12) NOT NULL,
customer_address VARCHAR2(255) NOT NULL
);
CREATE TABLE courier (
courier_id NUMBER PRIMARY KEY,
courier_web VARCHAR2(100) NOT NULL,
courier_name VARCHAR2(50) NOT NULL,
courier_phone VARCHAR2(12) NOT NULL
);
CREATE TABLE seller (
seller_id NUMBER PRIMARY KEY,
seller_address VARCHAR2(255) NOT NULL,
seller_phone VARCHAR2(25) NOT NULL,
seller_name VARCHAR2(128) NOT NULL,
seller_email VARCHAR2(254) NOT NULL
);
CREATE TABLE product_category (
category_id NUMBER PRIMARY KEY,
category_name VARCHAR2(100) NOT NULL
);
CREATE TABLE brand (
brand_id NUMBER PRIMARY KEY,
brand_name VARCHAR2(100) NOT NULL
);
CREATE TABLE payment (
payment_id NUMBER PRIMARY KEY,
payment_date DATE DEFAULT SYSDATE NOT NULL,
payment_type VARCHAR2(10) NOT NULL,
amount DECIMAL(10,2) NOT NULL, --should be same as cart(total_price)
customer_id NUMBER REFERENCES customer(customer_id) ON DELETE CASCADE
);
CREATE TABLE cart (
cart_id NUMBER PRIMARY KEY,
cart_status VARCHAR2(15) DEFAULT 'Processing',
total_price DECIMAL(10,2) NOT NULL,--should be equal to all products in the cart
[product(price) * contains_product(product_amount)]
courier_id NUMBER REFERENCES courier(courier_id) ON DELETE CASCADE,
payment_id NUMBER REFERENCES payment(payment_id) ON DELETE CASCADE,
service_level VARCHAR2(25) NOT NULL,
tracking_number VARCHAR2(40)
```

```sql
);
CREATE TABLE product (
product_id NUMBER PRIMARY KEY,
product_name VARCHAR2(100) NOT NULL,
product_count NUMBER NOT NULL CHECK (product_count >= 0),
price DECIMAL(10,2) NOT NULL CHECK (price > 0),
seller_id NUMBER REFERENCES seller(seller_id) ON DELETE CASCADE,
brand_id NUMBER REFERENCES brand(brand_id) ON DELETE CASCADE,
category_id NUMBER REFERENCES product_category(category_id) ON DELETE
CASCADE
);
CREATE TABLE contains_product(
product_id NUMBER REFERENCES product(product_id) ON DELETE CASCADE,
cart_id NUMBER REFERENCES cart(cart_id) ON DELETE CASCADE,
product_amount NUMBER CHECK (product_amount > 0), --this should be less than
product_count
PRIMARY KEY (product_id, cart_id)
);
/*inserting rows of mock data*/
--Customers:
INSERT INTO customer (customer_id, customer_name, customer_email, customer_phone,
customer_address) VALUES (1, 'J Weber', 'jweber@gmail.com', '123456789', '123 main street');
INSERT INTO customer (customer_id, customer_name, customer_email, customer_phone,
customer_address) VALUES (2, 'Ms Feeld', 'feeld@gmail.com', '555156189', '24 oak street');
INSERT INTO customer (customer_id, customer_name, customer_email, customer_phone,
customer_address) VALUES (3, 'Mr Saad', 'saad1@gmail.com', '555999848', '33 welesely rd');
INSERT INTO customer (customer_id, customer_name, customer_email, customer_phone,
customer_address) VALUES (4, 'Cashwan', 'cashwan@gmail.com', '555444888', '18 river rd');
INSERT INTO customer (customer_id, customer_name, customer_email, customer_phone,
customer_address) VALUES (5, 'Betty DeLile', 'bdlile@gmail.com', '555841599', '906 cam ave');
--Couriers:
INSERT INTO courier (courier_id, courier_name, courier_web, courier_phone ) VALUES (1,
'CanPost', 'canpost.ca', '180055555');
INSERT INTO courier (courier_id, courier_name, courier_web, courier_phone ) VALUES (2,
'Fedex', 'fedex.com', '18005556515');
INSERT INTO courier (courier_id, courier_name, courier_web, courier_phone ) VALUES (3,
'UPS', 'ups.com', '18005558954');
--Sellers:
INSERT INTO seller (seller_id, seller_name, seller_address, seller_phone, seller_email)
VALUES (1, 'Max Computers', '123 Max Ave', '1800555755', 'contact@maxcomp.com');
INSERT INTO seller (seller_id, seller_name, seller_address, seller_phone, seller_email)
VALUES (2, 'Park Vintage', 'Bud St', '9068526268', 'info@parkvintage.com');
INSERT INTO seller (seller_id, seller_name, seller_address, seller_phone, seller_email)
VALUES (3, 'Min Computers', '321 Min St', '815784658', 'contact@mincomp.com');
```

```sql
INSERT INTO seller (seller_id, seller_name, seller_address, seller_phone, seller_email)
VALUES (4, 'ToysRUs', '189 Toy Place', '18885558181', 'cs@toyrus.com');
INSERT INTO seller (seller_id, seller_name, seller_address, seller_phone, seller_email)
VALUES (5, 'ABC Books', '89 John St', '8159265484', 'abc@yahoo.com');
--Product Categories:
INSERT INTO product_category (category_id, category_name) VALUES (1, 'Computer');
INSERT INTO product_category (category_id, category_name) VALUES (2, 'Clothing');
INSERT INTO product_category (category_id, category_name) VALUES (3, 'Toys');
INSERT INTO product_category (category_id, category_name) VALUES (4, 'Books');
INSERT INTO product_category (category_id, category_name) VALUES (5, 'Games');
--Brands:
INSERT INTO brand (brand_id, brand_name) VALUES (1, 'Apple');
INSERT INTO brand (brand_id, brand_name) VALUES (2, 'Microsoft');
INSERT INTO brand (brand_id, brand_name) VALUES (3, 'Hot Wheels');
INSERT INTO brand (brand_id, brand_name) VALUES (4, 'Penguin Books');
INSERT INTO brand (brand_id, brand_name) VALUES (5, 'Nintendo');
INSERT INTO brand (brand_id, brand_name) VALUES (6, 'Vintage');
--Products:
INSERT INTO product (product_id, product_name, product_count, price, brand_id, category_id,
seller_id) VALUES (1, 'Apple Monitor', 20, 4000.00, 1, 1, 1);
INSERT INTO product (product_id, product_name, product_count, price, brand_id, category_id,
seller_id) VALUES (2, 'Microsoft Keyboard', 50, 75.00, 2, 1, 3);
INSERT INTO product (product_id, product_name, product_count, price, brand_id, category_id,
seller_id) VALUES (3, 'Hot Wheels Track Set', 15, 55.00, 3, 3, 4);
INSERT INTO product (product_id, product_name, product_count, price, brand_id, category_id,
seller_id) VALUES (4, 'Coffee Table Book', 20, 10.00, 4, 4, 5);
INSERT INTO product (product_id, product_name, product_count, price, brand_id, category_id,
seller_id) VALUES (5, 'Zelda', 40, 70.00, 5, 5, 4);
INSERT INTO product (product_id, product_name, product_count, price, brand_id, category_id,
seller_id) VALUES (6, 'Vintage 80s Military Jacket M', 10, 25.00, 6, 2, 2);
--Payments/Carts/Contains_Product:
INSERT INTO payment (payment_id, payment_date, payment_type, amount, customer_id)
VALUES (1, '2020-10-5', 'Visa', 4000.00, 1);
INSERT INTO cart (cart_id, cart_status, total_price, courier_id, payment_id, service_level,
tracking_number) VALUES (1, 'Shipped', 4000.00, 1, 1, 'Regular', 'xn151890op');
INSERT INTO contains_product (product_id, cart_id, product_amount) VALUES (1, 1, 1);
INSERT INTO payment (payment_id, payment_date, payment_type, amount, customer_id)
VALUES (2, '2020-10-20', 'MasterCard', 125.00, 2);
INSERT INTO cart (cart_id, cart_status, total_price, courier_id, payment_id, service_level,
tracking_number) VALUES (2, 'Processing', 125.00, 2, 2, 'Regular', '0605481028700');
INSERT INTO contains_product (product_id, cart_id, product_amount) VALUES (3, 2, 1);
INSERT INTO contains_product (product_id, cart_id, product_amount) VALUES (5, 2, 1);
INSERT INTO payment (payment_id, payment_date, payment_type, amount, customer_id)
VALUES (3, '2020-10-18', 'Interac', 70.00, 3);
```

```sql
INSERT INTO cart (cart_id, cart_status, total_price, courier_id, payment_id, service_level,
tracking_number) VALUES (3, 'Shipped', 70.00, 3, 3, 'Express', 'EX8189156');
INSERT INTO contains_product (product_id, cart_id, product_amount) VALUES (6, 3, 1);
INSERT INTO payment (payment_id, payment_date, payment_type, amount, customer_id)
VALUES (4, '2020-10-01', 'PayPal', 85.00, 4);
INSERT INTO cart (cart_id, cart_status, total_price, courier_id, payment_id, service_level,
tracking_number) VALUES (4, 'Shipped', 85.00, 1, 4, 'Xpress', 'X81865189EN');
INSERT INTO contains_product (product_id, cart_id, product_amount) VALUES (2, 4, 1);
INSERT INTO contains_product (product_id, cart_id, product_amount) VALUES (4, 4, 1);
/*queries*/
SELECT customer_name, customer_email
FROM customer
ORDER BY customer_name;
SELECT courier_name, courier_web, courier_phone
FROM courier
ORDER BY courier_name;
SELECT seller_name, seller_email, seller_phone, seller_address
FROM seller
ORDER BY seller_name;
SELECT category_name
FROM product_category
ORDER BY category_name;
SELECT brand_name
FROM brand
ORDER BY brand_name;
SELECT customer_name, amount, payment_type, payment_date
FROM customer, payment
WHERE customer.customer_id = payment.customer_id
ORDER BY payment_date;
SELECT product_name, product_count, price, seller_name
FROM product, seller
WHERE product.seller_id = seller.seller_id
ORDER BY product_name;
SELECT customer_name, cart_status, tracking_number
FROM customer, cart, payment
WHERE cart.payment_id = payment.payment_id
AND payment.customer_id = customer.customer_id
ORDER BY customer_name;
SELECT customer_name, product_name, product_amount
FROM customer, product, contains_product, cart, payment
WHERE contains_product.product_id = product.product_id
AND contains_product.cart_id = cart.cart_id
AND cart.payment_id = payment.payment_id
AND payment.customer_id = customer.customer_id
```

```sql
ORDER BY customer_name;
/*drop scripts*/
--SELECT 'DROP TABLE "' || TABLE_NAME || '" CASCADE CONSTRAINTS;' FROM
user_tables;


/*simple queries*/
SELECT customer_name, customer_email
FROM customer
ORDER BY customer_name;

SELECT courier_name, courier_web, courier_phone
FROM courier
ORDER BY courier_name;

SELECT seller_name, seller_email, seller_phone, seller_address
FROM seller
ORDER BY seller_name;

SELECT category_name
FROM product_category
ORDER BY category_name;

SELECT brand_name
FROM brand
ORDER BY brand_name;

/* join queries */
SELECT customer_name, amount, payment_type, payment_date
FROM customer, payment
WHERE customer.customer_id = payment.customer_id
ORDER BY payment_date;

SELECT product_name, product_count, price, seller_name
FROM product, seller
WHERE product.seller_id = seller.seller_id
ORDER BY product_name;

SELECT customer_name, cart_status, tracking_number
FROM customer, cart, payment
WHERE cart.payment_id = payment.payment_id
AND payment.customer_id = customer.customer_id
ORDER BY customer_name;
```

```sql
SELECT customer_name, product_name, product_amount
FROM customer, product, contains_product, cart, payment
WHERE contains_product.product_id = product.product_id
AND contains_product.cart_id = cart.cart_id
AND cart.payment_id = payment.payment_id
AND payment.customer_id = customer.customer_id
ORDER BY customer_name;

--Advanced join query
SELECT DISTINCT product_name, brand_name, category_name, seller_name
FROM product, product_category, brand, seller
WHERE product.brand_id = brand.brand_id
AND product.category_id = product_category.category_id
AND product.seller_id = seller.seller_id
ORDER BY product_name;
--Customers who bought Vintage 80s OR bought coffee table
SELECT customer_name
FROM customer, product, contains_product, cart, payment
WHERE contains_product.product_id = product.product_id
AND contains_product.cart_id = cart.cart_id
AND cart.payment_id = payment.payment_id
AND payment.customer_id = customer.customer_id
AND product_name = 'Vintage 80s Military Jacket M'
UNION
(SELECT customer_name
FROM customer, product, contains_product, cart, payment
WHERE contains_product.product_id = product.product_id
AND contains_product.cart_id = cart.cart_id
AND cart.payment_id = payment.payment_id
AND payment.customer_id = customer.customer_id
AND product_name = 'Coffee Table Book');
--Aggregate function, # of items sold
SELECT product_name, COUNT(*)
FROM contains_product, product
WHERE contains_product.product_id = product.product_id
GROUP BY product_name;
--Products that cost above $20 and $70 or under
SELECT product_name
FROM product
WHERE price BETWEEN 20 AND 70;
--Customers who have their products shipped by couriers other than Canpost
SELECT customer_name, cart_status
FROM customer, cart, payment
```

```sql
WHERE cart.payment_id = payment.payment_id
AND payment.customer_id = customer.customer_id
AND cart_status = 'Shipped'
AND NOT EXISTS
(SELECT *
FROM courier
WHERE courier_id = 1
AND cart.courier_id = courier.courier_id);
--Products sold by ToysRUs
SELECT product_name
FROM product
WHERE NOT EXISTS
(SELECT *
FROM seller
WHERE seller_id <> 4
AND product.seller_id = seller.seller_id);
```

```sql
--VIEWS
--Potential sale of product if you have a lot of extra inventory, amount of product is greater than
the average product count of all products
CREATE VIEW products_to_put_on_sale (discounted_productName,
discounted_productCount, original_price) AS
(SELECT product_name, product_count, price
FROM product
WHERE product_count>(
SELECT AVG(product_count)
FROM product));
```

```sql
--Customer that bought the most vintage jackets, could be customized to show all products top
shopper
CREATE VIEW VJ_Number_One_Shopper (Number_One_Customer) AS
SELECT customer_name
FROM customer, payment, cart, contains_product
WHERE product_amount =(
SELECT MAX(product_amount)
FROM contains_product
WHERE product_id = 6)
AND product_id = 6
AND contains_product.cart_id = cart.cart_id
AND cart.payment_id = payment.payment_id
AND payment.customer_id = customer.customer_id;
```

--Customer's who's orders are still being processed

CREATE VIEW need_to_be_shipped(cust_name, cust_address, track_num, sell_name, sell_phone)AS

SELECT DISTINCT customer_name, customer_address, tracking_number, seller_name, seller_phone

FROM customer, payment, cart, contains_product, product, seller

WHERE cart.payment_id = payment.payment_id

AND payment.customer_id = customer.customer_id

AND cart_status = 'Processing'
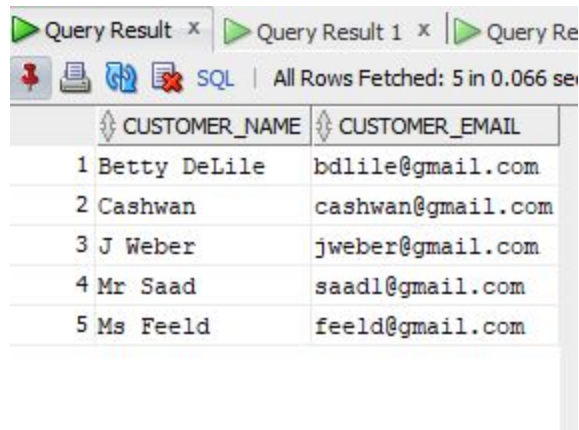
AND cart.cart_id = contains_product.cart_id

AND contains_product.product_id = product.product_id

AND product.seller_id = seller.seller_id

Query Results:

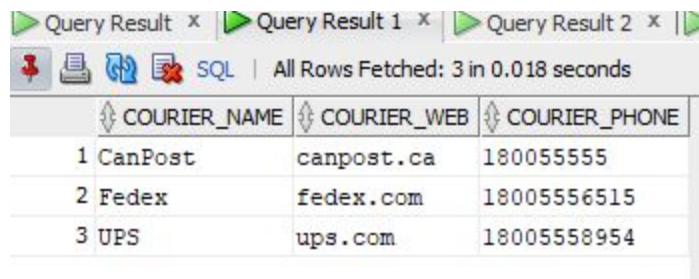Query Result × | Query Result 1 × | Query Re
SQL | All Rows Fetched: 5 in 0.066 se

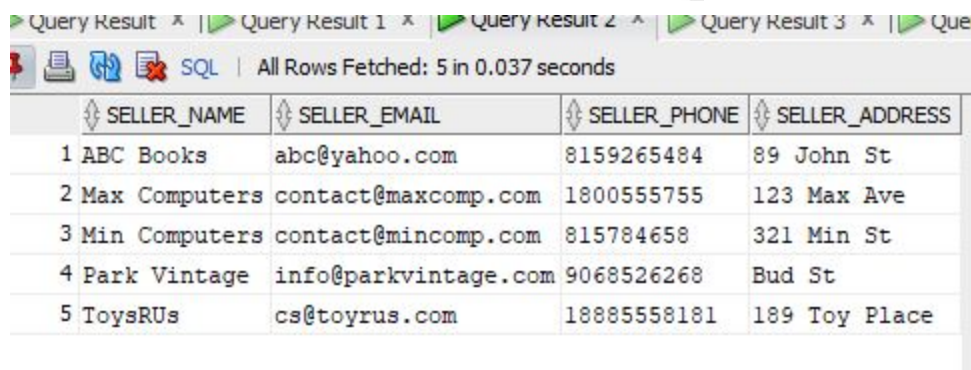|	CUSTOMER_NAME	CUSTOMER_EMAIL
1	Betty DeLile	bdlile@gmail.com
2	Cashwan	cashwan@gmail.com
3	J Weber	jweber@gmail.com
4	Mr Saad	saad1@gmail.com
5	Ms Feeld	feeld@gmail.com

Query Result × | Query Result 1 × | Query Result 2 × |
SQL | All Rows Fetched: 3 in 0.018 seconds

	COURIER_NAME	COURIER_WEB	COURIER_PHONE
1	CanPost	canpost.ca	180055555
2	Fedex	fedex.com	18005556515
3	UPS	ups.com	18005558954

Query Result × | Query Result 1 × | Query Result 2 × | Query Result 3 × | Que
SQL | All Rows Fetched: 5 in 0.037 seconds

	SELLER_NAME	SELLER_EMAIL	SELLER_PHONE	SELLER_ADDRESS
1	ABC Books	abc@yahoo.com	8159265484	89 John St
2	Max Computers	contact@maxcomp.com	1800555755	123 Max Ave
3	Min Computers	contact@mincomp.com	815784658	321 Min St
4	Park Vintage	info@parkvintage.com	9068526268	Bud St
5	ToysRUs	cs@toyrus.com	18885558181	189 Toy Place

	CATEGORY_NAME
1	Books
2	Clothing
3	Computer
4	Games
5	Toys

Query Result × | Query

SQL | All R

	BRAND_NAME
1	Apple
2	Hot Wheels
3	Microsoft
4	Nintendo
5	Penguin Books
6	Vintage

SQL | All Rows Fetched: 4 in 0.087 seconds

	CUSTOMER_NAME	AMOUNT	PAYMENT_TYPE	PAYMENT_DATE
1	Cashwan	85	PayPal	20-10-01
2	J Weber	4000	Visa	20-10-05
3	Mr Saad	70	Interac	20-10-18
4	Ms Feeld	125	MasterCard	20-10-20

	PRODUCT_NAME	PRODUCT_COUNT	PRICE	SELLER_NAME
1	Apple Monitor	20	4000	Max Computers
2	Coffee Table Book	20	10	ABC Books
3	Hot Wheels Track Set	15	55	ToysRUs
4	Microsoft Keyboard	50	75	Min Computers
5	Vintage 80s Military Jacket M	10	25	Park Vintage
6	Zelda	40	70	ToysRUs

Query Result × | Query Result 1 × | Query Result 2 × | Qu

SQL | All Rows Fetched: 4 in 0.055 seconds

	CUSTOMER_NAME	CART_STATUS	TRACKING_NUMBER
1	Cashwan	Shipped	X81865189EN
2	J Weber	Shipped	xn151890op
3	Mr Saad	Shipped	EX8189156
4	Ms Feeld	Processing	0605481028700

	CUSTOMER_NAME	PRODUCT_NAME	PRODUCT_AMOUNT
1	Cashwan	Microsoft Keyboard	1
2	Cashwan	Coffee Table Book	1
3	J Weber	Apple Monitor	1
4	Mr Saad	Vintage 80s Military Jacket M	1
5	Ms Feeld	Zelda	1
6	Ms Feeld	Hot Wheels Track Set	1