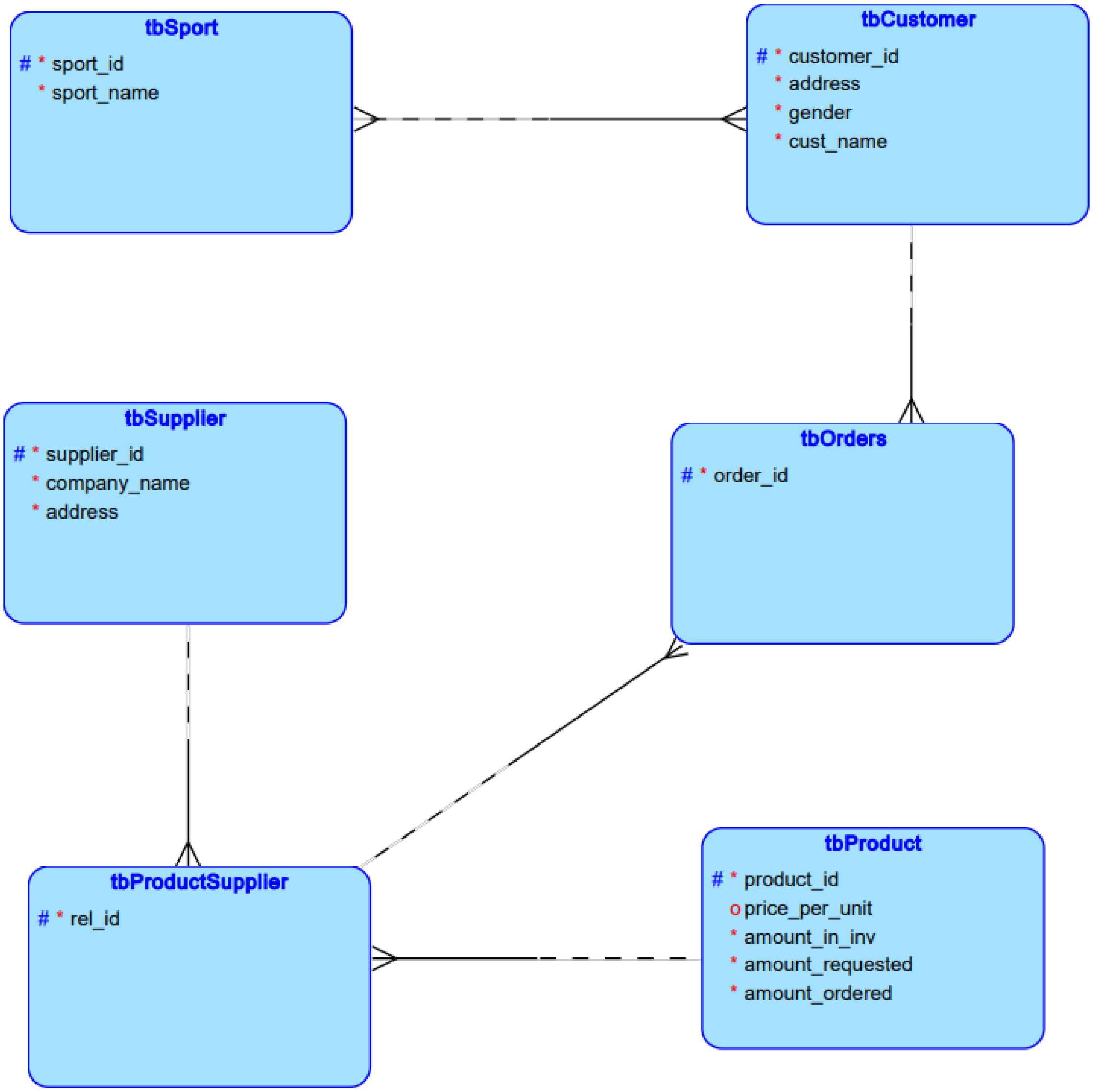
Kassem Faisal

Introduction to Databases.
Project:
Database for Memorabilia's Website
Task 5

1. Introduction

Memorabilia is an on-line company that buys sports products from various producers around the country and sells them to on-line customers. Customers visit Memorabilia’s Web site, select an item, and make an order. As soon as the customer's order is received, the product is delivered to the customer, and the inventory level is updated. The company orders a particular product from a supplier when the inventory level drops below a certain level. The company has decided to maintain a detailed database of the customers, suppliers, and products to manage the operations. The following information is stored in the database: a. Customer: name, address, gender, and preferred sport. b. Supplier: supplier identification number, name of the company, and address. c. Product: product identification number, price per unit, the amount in the inventory, amount requested of the suppliers but not yet received, amount ordered by the customers but not yet shipped. A customer may order one or more products, and a product may be ordered by one or more customers. Additionally, products may be provided by one or more suppliers, and a supplier may provide more than one product. Using this information, draw an E-R diagram for Memora-bilia’s database. Clearly state any assumptions made.



2. SQL

2.1. Create several database tables and insert the data

```
CREATE TABLE tbSport( sport_id NUMBER NOT NULL, PRIMARY KEY(sport_id),
sport_name VARCHAR(255) NOT NULL UNIQUE );
```

```
CREATE TABLE tbCustomer(
customer_id NUMBER NOT NULL,
PRIMARY KEY(customer_id),
address VARCHAR(255) NOT NULL,
gender CHAR(1) NOT NULL CHECK (gender = 'M' or gender = 'F'),
cust_name VARCHAR(255) NOT NULL CHECK(cust_name !='' AND
REGEXP_LIKE(cust_name,'^.+[:space:].+') AND REGEXP_COUNT(cust_name, ' ')=1)
);
```

```
/* We need to create this table due to setting up many-to-many relation, to allow customer pick
several preferable sports */
```

```
CREATE TABLE tbCustPrefSport(
pref_id NUMBER NOT NULL,
PRIMARY KEY(pref_id),
customer_id NUMBER NOT NULL,
FOREIGN KEY(customer_id) REFERENCES tbCustomer(customer_id),
sport_id NUMBER NOT NULL,
FOREIGN KEY(sport_id) REFERENCES tbSport(sport_id)
);

CREATE TABLE tbSupplier(
supplier_id NUMBER NOT NULL,
PRIMARY KEY(supplier_id),
company_name VARCHAR(255) NOT NULL UNIQUE,
address VARCHAR(255)
);

CREATE TABLE tbProduct(
product_id NUMBER NOT NULL,
PRIMARY KEY(product_id),
price_per_unit NUMBER(10,4) DEFAULT 0,
amount_in_inv NUMBER DEFAULT 0 CHECK(amount_in_inv >= 0),
amount_requested NUMBER DEFAULT 0,
amount_ordered NUMBER DEFAULT 0
);

/* we create this table to keep the DB normalized, also it allows to specify which product comes
from which supplier */

CREATE TABLE tbProductSupplier(
rel_id NUMBER NOT NULL,
PRIMARY KEY(rel_id),
product_id NUMBER NOT NULL,
supplier_id NUMBER NOT NULL,
FOREIGN KEY(product_id) REFERENCES tbProduct(product_id),
FOREIGN KEY(supplier_id) REFERENCES tbSupplier(supplier_id)
);

/* the Main table to track committed orders */

CREATE TABLE tbOrders(
order_id NUMBER NOT NULL,
PRIMARY KEY(order_id),
customer_id NUMBER NOT NULL,
prod_sup_id NUMBER NOT NULL,
FOREIGN KEY(customer_id) REFERENCES tbCustomer(customer_id),
FOREIGN KEY(prod_sup_id) REFERENCES tbProductSupplier(rel_id)
);

INSERT INTO tbSport(SPORT_ID, SPORT_NAME)
VALUES(1,'Bodybuilding');

INSERT INTO tbSport(SPORT_ID, SPORT_NAME)
VALUES(2, 'Tennis');
```

```
INSERT INTO tbSport(SPORT_ID, SPORT_NAME)
VALUES(3,'Baseball');

INSERT INTO tbSport(SPORT_ID, SPORT_NAME)
VALUES(4,'Boxing');

INSERT INTO tbSport(SPORT_ID, SPORT_NAME)
VALUES(5,'MMA');

INSERT INTO tbSport(SPORT_ID, SPORT_NAME)
VALUES(6,'Swimming');

INSERT INTO tbSport(SPORT_ID, SPORT_NAME)
VALUES(7,'Rugby');

INSERT INTO tbSport(SPORT_ID, SPORT_NAME)
VALUES(8,'Ice Skating');

INSERT INTO tbSport(SPORT_ID, SPORT_NAME)
VALUES(9,'Basketball');

INSERT INTO tbCustomer(CUSTOMER_ID,CUST_NAME,ADDRESS,GENDER)
VALUES(1, 'Faisal Kassem', 'Jana Matejki 21/23', 'M');

INSERT INTO tbCustomer(CUSTOMER_ID,CUST_NAME,ADDRESS,GENDER)
VALUES(2, 'Ruikang Leng', 'Jana Matejki 21/23', 'M');

INSERT INTO tbCustomer(CUSTOMER_ID,CUST_NAME,ADDRESS,GENDER)
VALUES(3, 'Mike Tyson', 'New York, Some Street 21/23', 'M');

INSERT INTO tbCustomer(CUSTOMER_ID,CUST_NAME,ADDRESS,GENDER)
VALUES(4, 'Connor McGregor', 'Ireland, Some Street 22/24', 'M');

INSERT INTO tbCustomer(CUSTOMER_ID, CUST_NAME, ADDRESS, GENDER)
VALUES(5, 'Rocky Balboa', 'New York, Some Street 25/54', 'M');

INSERT INTO tbCustomer(CUSTOMER_ID, CUST_NAME, ADDRESS, GENDER)
VALUES(6, 'Koby Briant', 'California, Some Street 25/22','M');

INSERT INTO tbCustomer(CUSTOMER_ID, CUST_NAME, ADDRESS, GENDER)
VALUES(7, 'Angelina Jolie', 'San Francisco, Some Street', 'F');

INSERT INTO tbCustPrefSport(PREF_ID, CUSTOMER_ID, SPORT_ID)
VALUES(1, 1, 1);

INSERT INTO tbCustPrefSport(PREF_ID, CUSTOMER_ID, SPORT_ID)
VALUES(2, 2, 5);

INSERT INTO tbCustPrefSport(PREF_ID, CUSTOMER_ID, SPORT_ID)
VALUES(3, 3, 4);

INSERT INTO tbCustPrefSport(PREF_ID, CUSTOMER_ID, SPORT_ID)
VALUES(4, 4, 5);

INSERT INTO tbCustPrefSport(PREF_ID, CUSTOMER_ID, SPORT_ID)
VALUES(5, 5, 7);
```

```
INSERT INTO tbCustPrefSport(PREF_ID, CUSTOMER_ID, SPORT_ID)
VALUES(6, 5, 1);

INSERT INTO tbCustPrefSport(PREF_ID, CUSTOMER_ID, SPORT_ID)
VALUES(7, 6, 9);

INSERT INTO tbCustPrefSport(PREF_ID, CUSTOMER_ID, SPORT_ID)
VALUES(8, 7, 2);

INSERT INTO tbSupplier(SUPPLIER_ID,COMPANY_NAME,ADDRESS)
VALUES(1,'Nike','One Bowerman Drive, Beaverton, OR 97005');

INSERT INTO tbSupplier(SUPPLIER_ID,COMPANY_NAME,ADDRESS)
VALUES(2,'GoldsGym','Dallas, Texas, SomeStreetOfGreats');

INSERT INTO tbSupplier(SUPPLIER_ID,COMPANY_NAME,ADDRESS)
VALUES(3,'Wilson', 'Chicago Illinois Street 22');

INSERT INTO tbSupplier(SUPPLIER_ID,COMPANY_NAME,ADDRESS)
VALUES(4,'Under Armour','Baltimore, Merilend Street');

INSERT INTO tbProduct(PRODUCT_ID, PRICE_PER_UNIT, AMOUNT_IN_INV,
AMOUNT_REQUESTED, AMOUNT_ORDERED)
VALUES(1, 100.50, 10, 0,0);

INSERT INTO tbProduct(PRODUCT_ID, PRICE_PER_UNIT, AMOUNT_IN_INV,
AMOUNT_REQUESTED, AMOUNT_ORDERED)
VALUES(2, 25, 20, 0,0);

INSERT INTO tbProduct(PRODUCT_ID, PRICE_PER_UNIT, AMOUNT_IN_INV,
AMOUNT_REQUESTED, AMOUNT_ORDERED)
VALUES(3, 34, 15, 0,0);

INSERT INTO tbProduct(PRODUCT_ID, PRICE_PER_UNIT, AMOUNT_IN_INV,
AMOUNT_REQUESTED, AMOUNT_ORDERED)
VALUES(4, 500, 5, 0,0);

INSERT INTO tbProduct(PRODUCT_ID, PRICE_PER_UNIT, AMOUNT_IN_INV,
AMOUNT_REQUESTED, AMOUNT_ORDERED)
VALUES(5, 200, 30, 0,0);

INSERT INTO tbProductSupplier(REL_ID, PRODUCT_ID, SUPPLIER_ID)
VALUES(1, 1, 4);

INSERT INTO tbProductSupplier(REL_ID, PRODUCT_ID, SUPPLIER_ID)
VALUES(2, 5, 2);

INSERT INTO tbProductSupplier(REL_ID, PRODUCT_ID, SUPPLIER_ID)
VALUES(3, 4, 3);

INSERT INTO tbProductSupplier(REL_ID, PRODUCT_ID, SUPPLIER_ID)
VALUES(4, 2, 1);

INSERT INTO tbProductSupplier(REL_ID, PRODUCT_ID, SUPPLIER_ID)
VALUES(5, 3, 1);
```

```
INSERT INTO tbOrders(ORDER_ID, CUSTOMER_ID, PROD_SUP_ID)
VALUES(1,1, 1);

UPDATE tbProduct
SET amount_in_inv = amount_in_inv-1
WHERE product_id = 1;

INSERT INTO tbOrders(ORDER_ID, CUSTOMER_ID, PROD_SUP_ID)
VALUES(2, 2, 4);

UPDATE tbProduct
SET amount_in_inv = amount_in_inv-1
WHERE product_id = 2;

INSERT INTO tbOrders(ORDER_ID, CUSTOMER_ID, PROD_SUP_ID)
VALUES(3, 3, 5);

UPDATE tbProduct
SET amount_in_inv = amount_in_inv-1
WHERE product_id = 4;

INSERT INTO tbOrders(ORDER_ID, CUSTOMER_ID, PROD_SUP_ID)
VALUES(4, 4, 2);

UPDATE tbProduct
SET amount_in_inv = amount_in_inv-1
WHERE product_id = 5;

INSERT INTO tbOrders(ORDER_ID, CUSTOMER_ID, PROD_SUP_ID)
VALUES(5, 5, 3);

UPDATE tbProduct
SET amount_in_inv = amount_in_inv-1
WHERE product_id = 4;

INSERT INTO tbOrders(ORDER_ID, CUSTOMER_ID, PROD_SUP_ID)
VALUES(6, 1, 5);

UPDATE tbProduct
SET amount_in_inv = amount_in_inv-1
WHERE product_id = 3;

INSERT INTO tbOrders(ORDER_ID, CUSTOMER_ID, PROD_SUP_ID)
VALUES(7, 7, 2);

UPDATE tbProduct
SET amount_in_inv = amount_in_inv-1
WHERE product_id = 5;

COMMIT WORK;
```

2.2. SELECT Queries

```
/* Num 1*/

SELECT CUSTOMER_ID ,
ADDRESS ,
GENDER ,
CUST_NAME FROM tbCustomer;

/* Num 2 ORDER BY */

SELECT COMPANY_NAME FROM tbSupplier
ORDER BY COMPANY_NAME ASC;

/* Num 3 MAX */

SELECT MAX(amount_in_inv) FROM tbProduct;

/* Num 4 Left JOIN + AGGR SUM /
SELECT s.COMPANY_NAME, SUM(pr.PRICE_PER_UNITpr.AMOUNT_IN_INV) FROM
tbSupplier s
LEFT JOIN tbProductSupplier ps ON s.SUPPLIER_ID = ps.SUPPLIER_ID
LEFT JOIN tbProduct pr ON pr.PRODUCT_ID = ps.PRODUCT_ID
GROUP BY(s.COMPANY_NAME);

/* Num 5 LEFT JOIN + AGGR AVG */

SELECT s.COMPANY_NAME, AVG(pr.PRICE_PER_UNIT) FROM
tbSupplier s
LEFT JOIN tbProductSupplier ps ON s.SUPPLIER_ID = ps.SUPPLIER_ID
LEFT JOIN tbProduct pr ON pr.PRODUCT_ID = ps.PRODUCT_ID
GROUP BY(s.COMPANY_NAME);

/* Num 6 HAVING USAGE */

SELECT sup.company_name, SUM(prd.price_per_unitprd.AMOUNT_IN_INV) FROM tbOrders
ord
JOIN tbProductSupplier psp ON psp.rel_id = ord.PROD_SUP_ID
JOIN tbProduct prd ON psp.product_id = prd.product_id
JOIN tbSupplier sup ON sup.supplier_id = psp.SUPPLIER_ID group by sup.company_name
HAVING SUM(prd.price_per_unitprd.AMOUNT_IN_INV) > (SELECT
AVG(price_per_unit*amount_in_inv) FROM tbProduct);

/* Num 7 COUNT */

SELECT COUNT(ord.customer_id), cus.cust_name FROM tbOrders ord
JOIN tbCustomer cus ON cus.customer_id = ord.customer_id
GROUP BY cus.cust_name ORDER BY 1 DESC;

/* Num 8 LIKE */

SELECT cust_name FROM tbCustomer
WHERE cust_name LIKE '%a%a%';

/* Num 9 WHERE+COUNT+Subquery+LIKE */
```

```sql
SELECT COUNT(*) FROM tbCustomer
WHERE customer_id IN (SELECT customer_id FROM tbCustomer WHERE gender = 'F' AND
cust_name LIKE 'A%');

/* Num 10 UNION */

SELECT supplier_id AS ID, company_name AS STR FROM tbSupplier
UNION
SELECT sport_id, sport_name FROM tbSport;

/* Num 11 INTERSECT */

SELECT sport_id FROM tbSport
INTERSECT
SELECT sport_id FROM tbCustPrefSport;

/* Num 12 Operation */

SELECT COUNT(prs.product_id) + prd.AMOUNT_IN_INV, prd.AMOUNT_IN_INV LeftItems
FROM tbOrders ord
JOIN tbProductSupplier prs ON ord.PROD_SUP_ID = prs.REL_ID
JOIN tbProduct prd ON prd.PRODUCT_ID = prs.PRODUCT_ID
GROUP BY prs.product_id, prd.AMOUNT_IN_INV;

/* Num 13 Between */

SELECT product_id, amount_in_inv FROM tbProduct
WHERE price_per_unit BETWEEN 125 AND 550;

/* Num 14 SUBSTR */

SELECT SUBSTR(cust_name,0,6) FROM tbCustomer WHERE customer_id = 1;

/* Num 15 DISTINCT */

SELECT DISTINCT price_per_unit UNIQPRICES FROM tbProduct;

/* Num 16 NESTED QUERY */

SELECT prd.product_id, (SELECT SUM(pr.price_per_unit)
FROM tbOrders ord LEFT JOIN tbProductSupplier ps ON ps.rel_id = ord.prod_sup_id
LEFT JOIN tbProduct pr ON ps.product_id = pr.product_id WHERE pr.product_id =
prd.product_id GROUP BY pr.product_id) AS TOTALSALES
FROM tbProduct prd;

/* Num 17 LIMIT */

SELECT * FROM (SELECT cust_name FROM tbCustomer ORDER BY 1 DESC) WHERE
ROWNUM<=3;

/* NUM 18 MINUS */

SELECT product_id, price_per_unit FROM tbProduct
MINUS
SELECT product_id, price_per_unit FROM tbProduct
WHERE PRICE_PER_UNIT < 100;

/* NUM 19 FULL ORDER DETAILS */
```

```
SELECT cus.cust_name, cus.ADDRESS, ps.product_id,sup.COMPANY_NAME, sup.address
FROM tbOrders ord
JOIN tbProductSupplier ps ON ps.rel_id = ord.PROD_SUP_ID
JOIN tbCustomer cus ON ord.customer_id = cus.customer_id
JOIN tbSupplier sup ON sup.SUPPLIER_ID = ps.SUPPLIER_ID;

/* NUM 20 WITH Keyword */

WITH all_orders AS(
SELECT COUNT(order_id) NumOfOrders, customer_id cid FROM tbOrders
GROUP BY customer_id
)
SELECT cus.cust_name, all_orders.NumOfOrders FROM all_orders, tbCustomer cus
WHERE cus.customer_id = all_orders.cid;
```

> It's like finding a needle in a huge pile of needles that lays in a massive haystack. The difference between theory and practice is that in theory there is no difference...