

**Database Systems Project**

**King’s Gym**

**Group (30)**

|  |  |
| --- | --- |
| **ID** | **Name** |
| 172739 | Omar Atef |
| 176311 | Omar Ahmed |
| 146607 | Mai Ezzat |
| 176468 | Hoda Ahmad |

**Contribution table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Mapping** | **Tables** | **Queries** |
| 172739 | Omar Atef | Member  Invitation  Member\_phone  Supplier  Supplies | Member  Invitation  Member\_phone  Supplier  Supplies | 6  7  8  9  10  Trigger Bonus |
| 176311 | Omar Ahmed | Exercise  Section  Participates  Expenses | Exercise  Section  Participates  Expenses | 1  2  3  4  5 |
| 146607 | Mai | Staff  Branch  StaffAddress  Has | Staff  Branch  StaffAddress  Has | 17  18  19  20  21 |
| 176468 | Hoda | Product  Equipment  Supplement  Coaches | Product  Equipment  Supplement  Coaches | 11  12  13  14,15  16 |

**Description of our Database:**

-The main topic is King's gym database. The King's gym database consists of 15 tables 4 of which are junction tables.

-Normal tables: Exercise, Member, Product, Supplement, Equipment, Staff, Expenses, Supplier, Section, Branch, Invitation.

-Junction tables: Participate, Supplies, Coaches, Has.

-Each table consists of attributes that describe/define the table’s content and each table also has a relationship with a different one in order to keep the database well-detailed and mutual.

**Functionality:**

**The goal of the database is to:**

* Store members’ data and the people they invited digitally instead of using paper to facilitate the retrieval of the information in and easy and quick way in order to offer them the facilities that match with the membership type or determine whether the membership is still valid.
* Organize the data of staff and divides them according to their hierarchies.
* List the transactions done between the gym and other entities (water/electricity companies) and individuals (like suppliers).
* Enclose the imports that are received by the gym during the transactions and labels each one of them with distinctive details
* Guide gym sessions by linking exercises to their continent/corresponding sections
* Allow members to purchase products from their branch store.

**Assumptions:**

**1.Exercise:**

-Each exercise is described by its name, duration, frequency per month and the part of the body it targets.

**2.Section:**

-The gym consists of 4 floors, a huge outdoor garden and two swimming pools (outdoor and indoor).

-The ground has a reception and a swimming pool, the first floor has upper body machines, the second floor has the main office and mid body machines, the third floor has lower body machines and the fourth floor has a spa.

-Not every section can be accessed by members or staff.

-Each exercising section on any floor is named after the exercise performed and exercises that don’t need equipment (aerobics) are performed in the garden.

**3.Expenses:**

-The gym deals with utility and maintenance expenses.

-Utility expenses such as water, electricity, gas, telephone and internet.

-Maintenance expenses such as equipment and utility check.

**4.Member:**

-Age restriction: 16 to 60.

-The first 2 digits of the members ID indicates they year he registered in the gym.

-The member can subscribe for a specific number of days.

-There are 3 types of memberships: Silver, Gold, Platinum.

**5.Member\_Phone:**

-A member can have more than 1 phone number registered.

**6.Supplier:**

-The suppliers are individuals. (Not working for a company)

-Each supplier supplies more than 1 different product.

**7.Invitation:**

-The invitation duration is for 1 day only.

-Only members with gold and platinum memberships can invite others.

-Each member can only invite 5 people.

**8.Supplements:**

-All supplements are nutrition-based products.

**9.Equipment:**

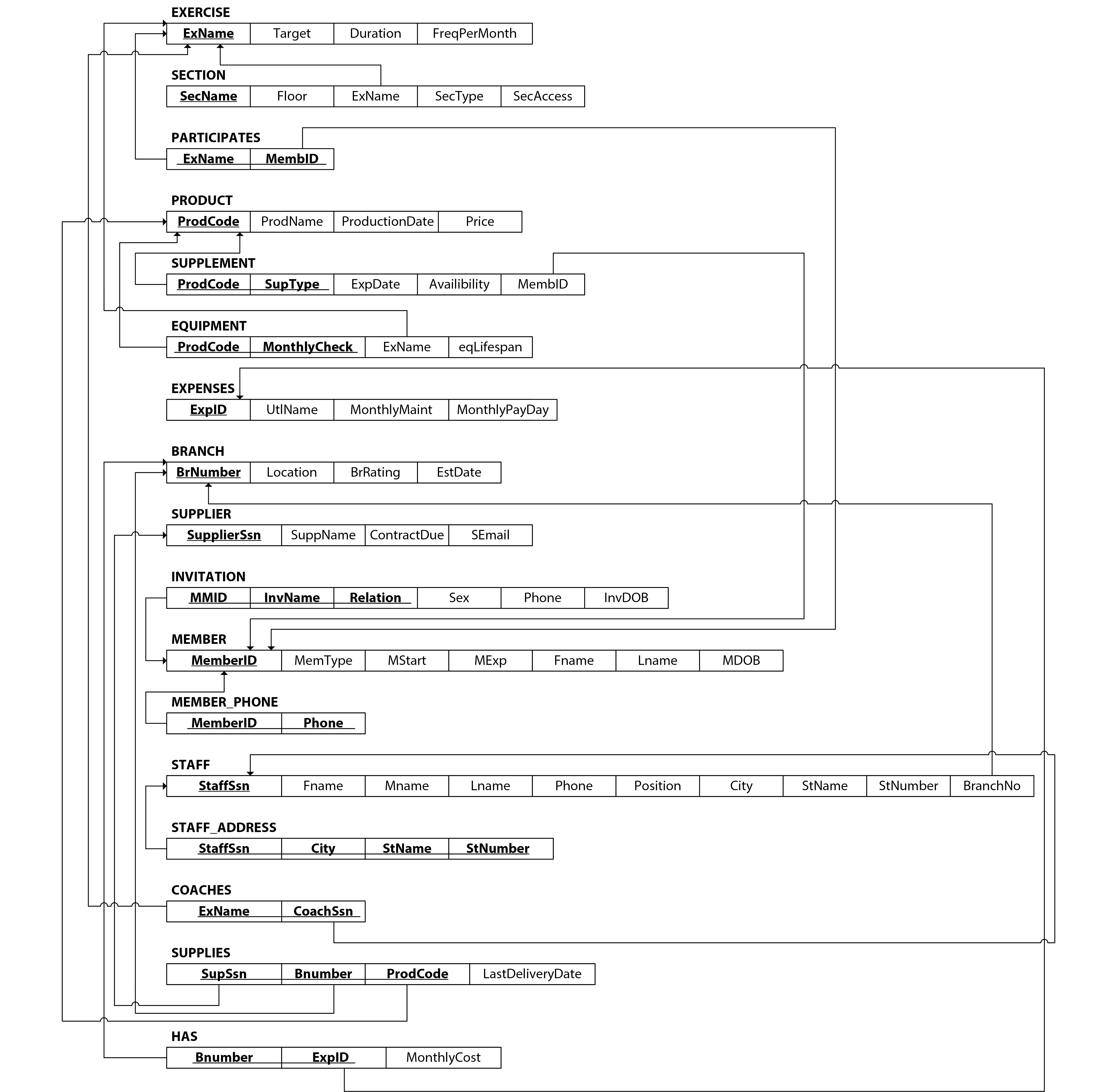
-The equipment is used by the member and bought by the gym.

**The Conceptual Schema (EERD):**

**A close up of a map

Description generated with high confidence**

**The Mapping / Logical Schema:**

****

**Code section:**

**Omar Ahmed 176311 Code:**

* **Tables**

CREATE TABLE Expenses

(

ExpID int not null,

UtlName varchar(15) not null,

MonthlyMaint int,

MonthlyPayDay int,

CONSTRAINT exp\_pk PRIMARY KEY (ExpID)

);

CREATE TABLE Exercise

(

ExName VARCHAR(20) NOT NULL,

Target VARCHAR(30) NOT NULL,

Duration VARCHAR(20),

FreqPerMonth INT

CONSTRAINT exe\_pk

PRIMARY KEY (ExName)

);

CREATE TABLE Participates

(

ExName VARCHAR(20) NOT NULL,

MembID int NOT NULL,

CONSTRAINT part\_pk

PRIMARY KEY (ExName,MembID),

CONSTRAINT exe\_part\_fk

FOREIGN KEY (ExName) REFERENCES Exercise(ExName),

CONSTRAINT memb\_part\_fk

FOREIGN KEY (MembID) REFERENCES Member(MemberID)

);

CREATE TABLE Section

(

SecName VARCHAR(25) NOT NULL,

Floor INT NOT NULL,

SecType VARCHAR(10),

SecAccess VARCHAR(30),

ExName VARCHAR(20),

CONSTRAINT sec\_pk

PRIMARY KEY (SecName),

CONSTRAINT sec\_fk

FOREIGN KEY (ExName) REFERENCES EXERCISE (ExName)

);

* **Inserted Data**

**Expenses table**

INSERT INTO Expenses

VALUES(1,'Water',1000,'1');

INSERT INTO Expenses

VALUES(2,'Electricity',2200,'8');

INSERT INTO Expenses

VALUES(3,'Telephone',750,'16');

INSERT INTO Expenses

VALUES(4,'Internet',1000,'12');

INSERT INTO Expenses

VALUES(5,'Gas',800,'3');

**Exercise table**

INSERT INTO Exercise

VALUES('Leg Press','Quadriceps','6 minutes',3);

INSERT INTO Exercise

VALUES('Bench Press','Chest','10 minutes',8);

INSERT INTO Exercise

VALUES('Pulldown','Mid Back','7 minutes',5);

INSERT INTO Exercise

VALUES('Upright Row','Shoulders','10 minutes',4);

INSERT INTO Exercise

VALUES('Push Ups','Chest','15 minutes',9);

INSERT INTO Exercise

VALUES('Tick-Top Hop','Quadriceps','15 minutes',3);

INSERT INTO Exercise

VALUES('Pushdown','Triceps','7 minutes',4);

INSERT INTO Exercise

VALUES('Preacher Curl','Biceps','8 minutes',5);

INSERT INTO Exercise

VALUES('Crunch','Waist','15 minutes',7);

INSERT INTO Exercise

VALUES('Chest fly','Chest','10 minutes',8);

INSERT INTO Exercise

VALUES('Military Press','Shoulders','20 minutes',6);

INSERT INTO Exercise

VALUES('Bicycle','Waist','10 minutes',2);

INSERT INTO Exercise

VALUES('Leg Curl','Hamstrings','6 minutes',5);

INSERT INTO Exercise

VALUES('Running','Lower Body','30 minutes',4);

**Participates table**

INSERT INTO Participates

VALUES('Preacher Curl',183000);

INSERT INTO Participates

VALUES('Leg Curl',183696);

INSERT INTO Participates

VALUES('Upright Row',184707);

INSERT INTO Participates

VALUES('Leg Press',172000);

INSERT INTO Participates

VALUES('Military Press',185893);

INSERT INTO Participates

VALUES('Chest fly',165255);

INSERT INTO Participates

VALUES('Crunch',161250);

INSERT INTO Participates

VALUES('Pulldown',173367);

INSERT INTO Participates

VALUES('Bench Press',181250);

INSERT INTO Participates

VALUES('Push Ups',187654);

INSERT INTO Participates

VALUES('Bicycle',165255);

INSERT INTO Participates

VALUES('Bicycle',173367);

INSERT INTO Participates

VALUES('Leg Press',181250);

**Section table**

INSERT INTO Section

VALUES('BenchPress Section',1,'Indoor','Member Only','Bench Press');

INSERT INTO Section

VALUES('Pushdown Section',1,'Indoor','Member Only','Pushdown');

INSERT INTO Section

VALUES('Garden 2',0,'Outdoor','All','Push Ups');

INSERT INTO Section

VALUES('Swimming Pool 1',0,'Indoor','All','Bicycle');

INSERT INTO Section

VALUES('LegPress',3,'Indoor','Member Only','Leg Press');

INSERT INTO Section

VALUES('Spa',4,'Indoor','Gold and Platinum Members',null);

INSERT INTO Section

VALUES('Swimming Pool 2',0,'Outdoor','All','Tick-Top Hop');

INSERT INTO Section

VALUES('PreacherCurl Section',2,'Indoor','Member Only','Preacher Curl');

INSERT INTO Section

VALUES('LegCurl Section',3,'Indoor','Member Only','Leg Curl');

INSERT INTO Section

VALUES('PullDown Section',2,'Indoor','Member Only','Pulldown');

INSERT INTO Section

VALUES('Garden 1',0,'Outdoor','All','Crunch');

INSERT INTO Section

VALUES('Main Office',2,'Indoor','Staff Only',null);

INSERT INTO Section

VALUES('ChestFly Section',1,'Indoor','Member Only','Chest Fly');

INSERT INTO Section

VALUES('Reception',0,'Indoor','All',null);

INSERT INTO Section

VALUES('Running Section',3,'Indoor','Member Only','Running');

INSERT INTO Section

VALUES('UprightRow Section',1,'Indoor','Member Only','Upright Row');

INSERT INTO Section

VALUES('MilitaryPress Section',1,'Indoor','Member Only','Military Press');

* **Queries**

1. Retrieve the expenses ID and utility name when the monthly maintenance cost is greater than 850.

SELECT ExpID,UtlName

FROM Expenses

WHERE MonthlyMaint>850;

A close up of a logo

Description generated with very high confidence

1. Retrieve the first and last name along with the member type of the members when the exercise performed targets the chest.

SELECT Fname+' '+Lname AS 'Full Name', MemType

FROM Member

WHERE MemberID IN (SELECT MembID

FROM Exercise, Participates

WHERE Participates.ExName=Exercise.ExName

AND Target = 'Chest');

A close up of a logo

Description generated with very high confidence

1. Retrieve the number of sections located on the first and fourth floor along with the floor it is located in.

SELECT count(SecName) AS 'Section Name' , Floor

FROM Section

WHERE Floor=1 OR Floor=4

GROUP BY Floor;

**A screenshot of a computer

Description generated with very high confidence**

1. Retrieve all exercise names, the body part it targets and duration along with the section it is performed in as long as the exercise is usually performed more than 5 times in a month.

SELECT SecName,Target, Exercise.ExName, Duration

FROM Exercise JOIN Section

ON Exercise.ExName=Section.ExName

WHERE FreqPerMonth>5;

A close up of a logo

Description generated with very high confidence

1. Retrieves all the section names, accessibility and the name of the exercise preformed in it.

SELECT SecName AS 'Section Name', SecAccess , Exercise.ExName

FROM Section LEFT OUTER JOIN Exercise

ON Section.ExName=Exercise.ExName;

A close up of a logo

Description generated with high confidence

**Omar Atef 172739** **Code:**

* **Tables**

CREATE TABLE Member

(

MemberID INT NOT NULL,

MemType VARCHAR(20) NOT NULL,

MStart VARCHAR(20) NOT NULL,

MExp VARCHAR(20) NOT NULL,

Fname VARCHAR(20) NOT NULL,

Lname VARCHAR(20) NOT NULL,

MDOB VARCHAR(20),

CONSTRAINT M\_pk PRIMARY KEY(MemberID)

);

CREATE TABLE Member\_Phone

(

MemberID INT NOT NULL,

Phone VARCHAR(20) NOT NULL,

CONSTRAINT Phone\_pk PRIMARY KEY (MemberID,Phone),

CONSTRAINT Phone\_fk FOREIGN KEY (MemberID) REFERENCES Member(MemberID)

ON UPDATE CASCADE

);

CREATE TABLE Invitation

(

MMID INT NOT NULL,

InvName VARCHAR(30),

Relation VARCHAR(20),

Sex CHAR,

Phone VARCHAR(20),

InvDOB VARCHAR(20),

CONSTRAINT Inv\_pk PRIMARY KEY (MMID,InvName,Relation),

CONSTRAINT Inv\_fk FOREIGN KEY (MMID) REFERENCES Member(MemberID)

ON UPDATE CASCADE

);

CREATE TABLE Supplier

(

SupplierSsn BIGINT NOT NULL,

SuppName VARCHAR(50),

ContractDue VARCHAR(20),

SEmail VARCHAR(30),

CONSTRAINT Sup\_pk PRIMARY KEY (SupplierSsn) );

CREATE TABLE Supplies

(

SupSsn BIGINT NOT NULL,

Bnumber INT,

ProdCode INT,

LastDeliveryDate VARCHAR(15)

CONSTRAINT supplies\_pk

PRIMARY KEY (SupSsn,Bnumber,ProdCode),

CONSTRAINT Ssnsup\_fk

FOREIGN KEY (SupSsn) REFERENCES Supplier(SupplierSsn),

CONSTRAINT bnum\_fk

FOREIGN KEY (Bnumber) REFERENCES Branch(B\_number),

CONSTRAINT pcode\_fk

FOREIGN KEY (ProdCode) REFERENCES Product(ProdCode)

);

* **Inserted Data**

**Member table**

INSERT INTO Member

VALUES('183000','Gold','22/10/2018','27/11/2018','Omar','ElGedawy','16/4/1999');

INSERT INTO Member

VALUES('183696','Silver','13/5/2018','7/8/2018','Moaz','Taha','20/5/1980');

INSERT INTO Member

VALUES('184707','Gold','16/4/2018','6/7/2018','John','Ayman','30/6/1990');

INSERT INTO Member

VALUES('172000','Platinum','15/3/2017','30/3/2017','Fayrouz','Ramzy','21/3/1987');

INSERT INTO Member

VALUES('176097','Gold','11/2/2017','8/3/2017','Ann','Amin','15/7/1998');

INSERT INTO Member

VALUES('185893','Platinum','19/1/2018','10/5/2018','Mohamed','Radwan','8/9/1996');

INSERT INTO Member

VALUES('165255','Silver','17/8/2016','11/9/2016','Zyad','ElManzalawy','17/12/1957');

INSERT INTO Member

VALUES('161250','Gold','9/6/2016','14/9/2016','Raafat','Medhat','3/4/1969');

INSERT INTO Member

VALUES('173367','Gold','15/4/2017','16/8/2017','Belal','ElAraby','13/9/1988');

INSERT INTO Member

VALUES('181250','Platinum','19/6/2018','7/12/2018','Mahmoud','Magdy','3/4/1997');

INSERT INTO Member

VALUES('187654','Platinum','27/1/2018','27/1/2019','Rami','Helal','23/10/1970');

INSERT INTO Member

VALUES('183344','Platinum','24/12/2018','3/5/2019','Ahmed','Khaled','2/4/1993');

INSERT INTO Member

VALUES('182200','Gold','4/8/2018','13/9/2018','Monem','Marzouk','2/12/1990');

INSERT INTO Member

VALUES('188888','Silver','26/2/2018','25/4/2018','AbdelKhalek','Awad','2/4/1993');

INSERT INTO Member

VALUES('187676','Gold','20/2/2018','28/9/2018','Ahmed','Awad',NULL);

INSERT INTO Member

VALUES('186632','Silver','10/7/2018','28/12/2018','Tarek','ElZeeny',NULL);

INSERT INTO Member

VALUES('184451','Platinum','23/7/2018','17/11/2018','Omar','Karam',NULL);

**Member\_Phone Table**

INSERT INTO Member\_Phone

VALUES ('183000','+00201147108923');

INSERT INTO Member\_Phone

VALUES ('183000','+00201141892100');

INSERT INTO Member\_Phone

VALUES ('184707','+00201158784321');

INSERT INTO Member\_Phone

VALUES ('184707','+00201248444651');

INSERT INTO Member\_Phone

VALUES ('172000','+00201138886610');

INSERT INTO Member\_Phone

VALUES ('172000','+00201149797899');

INSERT INTO Member\_Phone

VALUES ('165255','+00201148107843');

INSERT INTO Member\_Phone

VALUES ('181250','+00201154448731');

INSERT INTO Member\_Phone

VALUES ('173367','+00201148765321');

INSERT INTO Member\_Phone

VALUES ('161250','+00201123365439');

**Invitation Table**

INSERT INTO Invitation

VALUES('183000','Osama Shawky','Cousin','M','01134978943','25/3/1996');

INSERT INTO Invitation

VALUES('183000','Mariam Sameh','Friend','F','0118888946','17/5/1990');

INSERT INTO Invitation

VALUES('184707','Marina Emad','Wife','F','01147772323','1/2/1992');

INSERT INTO Invitation

VALUES('172000','Menna Fouad','Friend','F','01089895553','10/5/1988');

INSERT INTO Invitation

VALUES('176097','Mark Samuel','Friend','M','01245431199','13/8/1996');

INSERT INTO Invitation

VALUES('185893','Kareem Radwan','Brother','M','01002894678','14/5/1993');

INSERT INTO Invitation

VALUES('185893','Mona Radwan','Sister','F','01276764533','9/7/1998');

INSERT INTO Invitation

VALUES('181250','Adel Hossam','Friend','M','01298652133','13/4/1994');

INSERT INTO Invitation

VALUES('181250','Amin ElShenawy','Cousin','M','01566798331','28/6/1996');

INSERT INTO Invitation

VALUES('173367','Yasmeen Rashwan','Wife','F','01377763421','24/7/1993');

INSERT INTO Invitation

VALUES('187654','Mayar Helal','Sister','F','01145533331','25/3/1975');

INSERT INTO Invitation

VALUES('187654','Mark Daniel','Friend','M','01157878913','9/7/1974');

**Supplier Table**

INSERT INTO Supplier

VALUES ('8786545634','Ahmed Farouk','22/10/2020','ahmedfarouk@yahoo.com');

INSERT INTO Supplier

VALUES ('6754376892','Reda Kamal','1/2/2021','redakamal@yahoo.com');

INSERT INTO Supplier

VALUES ('9757634921','Yasser Mortada','1/10/2019','yassermortada@yahoo.com');

INSERT INTO Supplier

VALUES ('4567342133','Marina Pola','28/12/2024','marinapola@gmail.com');

INSERT INTO Supplier

VALUES ('2434976523','Tahseen Ahmed','13/2/2022','tahseenahmed@hotmail.com');

INSERT INTO Supplier

VALUES ('9997675643','Amir Hesham','16/7/2019','amirhesham123@yahoo.com');

INSERT INTO Supplier

VALUES ('7765453233','Momen Ashraf','6/8/2025','momenashroof@gmail.com');

INSERT INTO Supplier

VALUES ('9874321785','Yasser ELMasry','13/2/2020','yasserelmasry@gmail.com');

INSERT INTO Supplier

VALUES ('9832118623','Michael Hanna','5/3/2021','mickieHHH@hotmail.com');

INSERT INTO Supplier

VALUES ('8647634297','Asser Mekawy','13/12/2019','mekawyasser33@gmail.com');

INSERT INTO Supplier

VALUES ('9856663491','Ashraf Gamal','1/8/2023','ashrafgmalservice@yahoo.com');

**Supplies Table**

INSERT INTO Supplies

VALUES ('9997675643', '1','100004','24/5/2018');

INSERT INTO Supplies

VALUES ('8786545634', '1','100000','31/1/2018');

INSERT INTO Supplies

VALUES ('6754376892', '2','100001','7/11/2018');

INSERT INTO Supplies

VALUES ('9757634921', '2','100002','6/12/2018');

INSERT INTO Supplies

VALUES ('4567342133', '3','100005','15/7/2018');

INSERT INTO Supplies

VALUES ('2434976523', '3','100006','28/2/2018');

INSERT INTO Supplies

VALUES ('7765453233', '3','100007','18/4/2018');

INSERT INTO Supplies

VALUES ('9874321785', '4','100008','9/3/2018');

INSERT INTO Supplies

VALUES ('9832118623', '4','100009','2/9/2018');

INSERT INTO Supplies

VALUES ('8647634297', '4','100010','20/10/2018');

INSERT INTO Supplies

VALUES ('9856663491', '4','100011','23/10/2018');

INSERT INTO Supplies

VALUES ('8786545634', '5','100004','12/8/2018');

INSERT INTO Supplies

VALUES ('4567342133', '5','100012','27/4/2018');

INSERT INTO Supplies

VALUES ('9856663491', '1','100013','27/4/2018');

* **Queries**

1. Retrieve the first and last names concatenated as “Full name” and IDs of the members that registered in 2018, have a platinum membership, and have the letter ‘a’ in the middle their first names.

SELECT Fname+' '+Lname AS 'Full Name', MemberID

FROM Member

WHERE MemberID>180000 AND MemType = 'Platinum' AND Fname LIKE '%a%';

A close up of a logo

Description generated with very high confidence

1. Retrieve the number of invitations grouped according to each member first and last names, but only retrieve the names of members who gave more than 1 invitations.



1. Retrieve the phone numbers of members who have a gold membership type. If a member hasn’t added any phones for contact, null values will be returned for the phones.

SELECT Fname+' '+Lname AS 'Name',Phone

FROM Member LEFT OUTER JOIN Member\_Phone

ON Member.MemberID=Member\_Phone.MemberID

WHERE MemType='Gold';

**A close up of a piece of paper

Description generated with high confidence**

1. Retrieve the suppliers’ emails and social security number who supply Omega Three Bars.

SELECT SEmail , SupplierSsn

FROM Supplier

WHERE SupplierSsn

In( SELECT SupSsn

FROM Supplies, Product

WHERE Supplies.ProdCode = Product.prodCode

AND ProdName = 'Omega Three Bars'

);

A close up of a logo

Description generated with very high confidence

1. Retrieve the first name, phone numbers and the dates of birth of all members and include phone numbers that don’t have a date of birth assigned to them and include dates of birth that don’t have phones numbers assigned to them.

SELECT Fname,Lname,MDOB,Phone

FROM Member FULL OUTER JOIN Member\_Phone

ON Member.MemberID = Member\_Phone.MemberID;

**A close up of a logo

Description generated with high confidence**

**Hoda 1764648 Code:**

* **Tables**

CREATE TABLE Product (

prodCode INT,

prodName VARCHAR(30) NOT NULL,

productionDate DATE,

price INT NOT NULL

CONSTRAINT prod\_pk

PRIMARY KEY (prodCode)

);

CREATE TABLE Supplement (

prodCode INT NOT NULL,

availability VARCHAR(1) NOT NULL,

expDate DATE,

supType VARCHAR(25) NOT NULL,

membID INT NOT NULL

CONSTRAINT supp\_pk

PRIMARY KEY (prodCode, SupType),

CONSTRAINT prodSup\_fk

FOREIGN KEY (prodCode) REFERENCES Product(prodCode)

ON DELETE CASCADE,

CONSTRAINT membSup\_fk

FOREIGN KEY (membID) REFERENCES Member(MemberID)

ON UPDATE CASCADE

);

CREATE TABLE Equipment (

prodCode INT NOT NULL,

monthlyCheck DATE NOT NULL,

exeName VARCHAR(20) NOT NULL,

eqLifespan VARCHAR(20)

CONSTRAINT equip\_pk

PRIMARY KEY (prodCode, monthlyCheck),

CONSTRAINT prodEquip\_fk

FOREIGN KEY (prodCode) REFERENCES Product(prodCode)

ON UPDATE CASCADE,

CONSTRAINT equipEx\_fk

FOREIGN KEY (exeName) REFERENCES Exercise(ExName)

ON UPDATE CASCADE

);

CREATE TABLE Coaches (

ExerName VARCHAR(20) NOT NULL,

coachSSN VARCHAR(15) NOT NULL

CONSTRAINT coaches\_pk

PRIMARY KEY (ExerName, coachSSN),

CONSTRAINT coach\_ex\_fk

FOREIGN KEY (ExerName) REFERENCES Exercise(ExName),

CONSTRAINT coach\_staff\_fk

FOREIGN KEY (coachSSN) REFERENCES staff(Ssn)

);

* **Inserted Data**

**Product Table**

INSERT INTO Product

VALUES ('100000','Multivitamin Bars','2018-1-1','20');

INSERT INTO Product

VALUES ('100001','Plates/Dumbbells','2011-1-5','500');

INSERT INTO Product

VALUES ('100002','Pro Amino','2018-1-15','800');

INSERT INTO Product

VALUES ('100003','Treadmill','2017-10-10','4500');

INSERT INTO Product

VALUES ('100004','Omega Three Bars','2018-12-29','120');

INSERT INTO Product

VALUES ('100005','Adjustable Bench','2018-10-20','700');

INSERT INTO Product

VALUES ('100006','Elite Creatine','2016-12-30','1250');

INSERT INTO Product

VALUES ('100007','Pull Down Machine','2017-1-1','2800');

INSERT INTO Product

VALUES ('100008','Amino Booster','2017-11-30','999');

INSERT INTO Product

VALUES ('100009','Preacher Curl Machine','2016-12-12','950');

INSERT INTO Product

VALUES ('100010','Builders Protien','2018-2-27','210');

INSERT INTO Product

VALUES ('100011','Plates/Dumbbells','2018-8-9','525');

INSERT INTO Product

VALUES ('100012','Elite Burn','2017-11-25','660');

INSERT INTO Product

VALUES ('100013','Leg Press Machine','2016-12-29','6000');

INSERT INTO Product

VALUES ('100014', 'Elite Protien', '2018-1-1', '15');

**Supplement Table**

INSERT INTO Supplement

VALUES ('100000','Y','2020-12-30','Vitamin','183000');

INSERT INTO Supplement

VALUES ('100002','Y','2022-12-30','Whey Protien','172000');

INSERT INTO Supplement

VALUES ('100004','N','2023-1-1','Vitamin','181250');

INSERT INTO Supplement

VALUES ('100006','N','2019-1-30','Creatine','187654');

INSERT INTO Supplement

VALUES ('100008','Y','2018-12-30','Whey Protien','161250');

INSERT INTO Supplement

VALUES ('100010','Y','2021-9-21','Protien Bar','184451');

INSERT INTO Supplement

VALUES ('100012','N','2020-12-30','Fat Burner','184707');

INSERT INTO Supplement

VALUES ('100014','Y','2018-12-12','Protien Drink', '184451');

**Equipment Table**

INSERT INTO Equipment

VALUES ('100001','2018-1-11','Military Press','10 Years');

INSERT INTO Equipment

VALUES ('100003','2018-2-12','Running','7 Months');

INSERT INTO Equipment

VALUES ('100005','2019-1-20','Bench Press','6 Months');

INSERT INTO Equipment

VALUES ('100007','2018-10-15','PullDown','2 Years');

INSERT INTO Equipment

VALUES ('100009','2018-5-20','Preacher Curl','1 Year');

INSERT INTO Equipment

VALUES ('100011','2018-10-21','Upright Row','10 months');

INSERT INTO Equipment

VALUES ('100013','2018-9-8','Leg Press','2 Years');

**Coaches Table**

INSERT INTO Coaches

VALUES ('Leg Press','165018977');

INSERT INTO Coaches

VALUES ('Bench Press','165018977');

INSERT INTO Coaches

VALUES ('Tick-Top Hop','235999363');

INSERT INTO Coaches

VALUES ('Chest fly','243648864');

INSERT INTO Coaches

VALUES ('Preacher Curl','243648864');

INSERT INTO Coaches

VALUES ('Upright Row','243648864');

INSERT INTO Coaches

VALUES ('Military Press','235999363');

INSERT INTO Coaches

VALUES ('Bicycle','235999363');

INSERT INTO Coaches

VALUES ('Leg Curl','165018977');

* **Queries**

1. This query retrieves ONLY the available supplements and their prices.

SELECT prodName AS 'Product Name', price, availability

FROM Product JOIN Supplement

ON Product.prodCode = Supplement.prodCode

WHERE availability = 'Y';

**A close up of a logo

Description generated with very high confidence**

1. This query retrieves the name and expiry date of the product(s) bought by all members whose first name is Omar.

SELECT prodName AS 'Product Name', expDate AS 'Expiry date',

membID AS 'Member ID'

FROM product join Supplement

ON Product.prodCode=Supplement.prodCode

WHERE membID IN

( SELECT memberID

FROM Member

WHERE membID = MemberID

AND Fname = 'omar'

);

A close up of a logo

Description generated with very high confidence

1. This Query retrieves the first names and IDs of all the members who made supplement purchases and the product code of what they purchased.

SELECT Fname + ' ' + Lname AS 'Name', MemberID AS 'Member ID', prodCode AS 'Product Code'

FROM Member JOIN Supplement

ON MemberID = membID

WHERE exists

(SELECT \*

FROM Supplement

WHERE MemberID = membID );

**A close up of a logo

Description generated with very high confidence**

1. Retrieves the supplement name, type, price, and the expiry date of supplements that their maximum prices are less than 500.

SELECT price, supType AS 'Supplement Type', expDate AS 'Expiry Date'

FROM Product JOIN Supplement

ON Product.prodCode = supplement.prodCode

GROUP BY price, supType, expDate

HAVING MAX(price) < 500;

A close up of a logo

Description generated with very high confidence

1. Retrieves the coach’ s name, the exercise he is responsible of, its target and duration. The coach’s first name must contain a second letter ‘o’, regardless of what comes afterwards.

SELECT fname + ' ' + lname AS 'Coach name', exName AS 'Exercise name', Target, duration

FROM Staff, Exercise

WHERE ExName

IN ( SELECT ExerName

FROM Coaches

WHERE coachSSN = ssn

)

AND fname LIKE '\_o%';

A close up of a logo

Description generated with very high confidence

1. Retrieves the product name, its price, the lifespan of the equipment and its monthly check for the products with a Minimum price that is more than 700.

SELECT prodName AS 'Product Name', price, eqLifespan AS 'Equipment lifespan', monthlyCheck

FROM Product FULL OUTER JOIN Equipment

ON Product.prodCode = Equipment.prodCode

GROUP BY prodName, eqLifespan, monthlyCheck, price

HAVING MIN(price) > 700;

A close up of a logo

Description generated with very high confidence

**Mai 146607** **Code:**

* **Tables**

CREATE TABLE Staff

( Fname Varchar(15) Not Null,

Minit Char,

Lname Varchar(15) Not Null,

Ssn VARCHAR(15) Not Null,

Phone INT,

Position Varchar(25),

Bnumber INT Not Null,

CONSTRAINT staff\_pk

PRIMARY KEY (Ssn),

CONSTRAINT branch\_fk

FOREIGN KEY (Bnumber) REFERENCES Branch(B\_number)

);

CREATE TABLE StaffAddress

( Ssn Varchar(15) Not Null,

City Varchar(15) Not Null,

PostalCode INT Not Null,

StName Varchar(15) Not Null,

CONSTRAINT staffAdd\_pk

PRIMARY KEY (Ssn, City, PostalCode, StName),

CONSTRAINT staffAdd\_fk

FOREIGN KEY (Ssn) REFERENCES Staff(Ssn)

);

CREATE TABLE Branch

( B\_number INT Not Null,

Location VARCHAR(15),

B\_rating Decimal(10,2),

EstDate VARCHAR(20)

CONSTRAINT branch\_pk

PRIMARY KEY (B\_number),

);

CREATE TABLE Has

( Bnumber INT Not Null,

ExpID INT Not Null,

MonthlyCost INT,

CONSTRAINT has\_pk

primary key (Bnumber, ExpID),

CONSTRAINT has\_fk1

foreign key (Bnumber) REFERENCES Branch(B\_number),

CONSTRAINT has\_fk2

foreign key (ExpID) REFERENCES Expenses(ExpID),

);

* **Inserted Data**

**Staff Table**

Insert into Staff

Values ('Gunther', 'C', 'Perk', '154336574', '+2010032468','Branch Manager','3')

Insert into Staff

Values('Rachel','K','Green','253836633','+2011502720','Receptionist','4')

Insert into Staff

Values('Ross','E','Geller','163828899','+2011285766','CEO','2')

Insert into Staff

Values ('Monica','E','Geller','243648864','+2010005005','Personal Trainer','5')

Insert into Staff

Values('Chandler','M','Bing','167491211','+2010103000','Accountant','5')

Insert into Staff

Values('Phoebe','H','Buffay','235999363','+2011426268','Trainer','1')

Insert into Staff

Values('Joey','F','Tribbiani','165018977','+2010003330','Trainer','1')

**StaffAddress Table**

Insert into StaffAddress

Values('154336574','Cairo','11765','Nasr City')

Insert into StaffAddress

Values('154336574','Giza','12611','Dokki')

Insert into StaffAddress

Values('253836633','Cairo','11757','Heliopolis')

Insert into StaffAddress

Values('253836633','Cairo','11841','Rehab City')

Insert into StaffAddress

Values('163828899','Giza','12588','Sheikh Zayed')

Insert into StaffAddress

Values('243648864','Cairo','11837','ElShrouk City')

Insert into StaffAddress

Values('167491211','Cairo','11799','Sheraton')

Insert into StaffAddress

Values('235999363','Giza','12655','AlMohandseen')

Insert into StaffAddress

Values('235999363','Giza','12566','6th October')

Insert into StaffAddress

Values('165018977','Giza','11568','Zamalek')

Insert into StaffAddress

Values('165018977','Cairo','11835','5th Settlement')

**Branch Table**

Insert into Branch

Values('1','Zamalek','9.5','30-6-1997')

Insert into Branch

Values('2','6th October','8.3','11-11-2009')

Insert into Branch

Values('3','Nasr City','8.0','2-2-2002')

Insert into Branch

Values('4','5th Settlement','8.8','')

Insert into Branch

Values('5','Maadi','9.2','15-8-2017')

**Has Table**

Insert into Has

Values('1','1','2500')

Insert into Has

Values('1','2','3000')

Insert into Has

Values('1','3','1000')

Insert into Has

Values('1','4','500')

Insert into Has

Values('1','5','800')

Insert into Has

Values('2','1','2200')

Insert into Has

Values('2','2','2500')

Insert into Has

Values('2','3','1100')

Insert into Has

Values('2','4','400')

Insert into Has

Values('2','5','900')

Insert into Has

Values('3','1','2100')

Insert into Has

Values('3','2','2000')

Insert into Has

Values('3','3','1200')

Insert into Has

Values('3','4','350')

Insert into Has

Values('3','5','600')

Insert into Has

Values('4','1','2300')

Insert into Has

Values('4','2','2700')

Insert into Has

Values('4','3','700')

Insert into Has

Values('4','5','1000')

Insert into Has

Values('5','1','2000')

Insert into Has

Values('5','2','2300')

Insert into Has

Values('5','3','1150')

Insert into Has

Values('5','5','650')

* **Queries**

1. Retrieves first name, middle initial, last name of staff as “Employee name”, and the branch numbers they work in.

SELECT Fname+' '+Minit+'. '+Lname AS 'Employee name',Bnumber

FROM Staff RIGHT OUTER JOIN Branch

ON Staff.Bnumber = Branch.B\_number;

A close up of a logo

Description generated with very high confidence

1. Retrieves the branch number, its location and the branch rating for those branches with a minimum rating that’s more than 8.5.

SELECT B\_number, Location, MIN(B\_rating) AS 'Branch Rating'

FROM Branch

Group by B\_number, Location

Having MIN(B\_rating) > 8.5;

A close up of a device

Description generated with high confidence

1. Retrieves staff’s first and last name as “full name”, street name and city as “Address” for those whose last names start with the letter ‘G’.

SELECT Fname +' '+ Lname AS 'Full Name', StName +', '+ City AS 'Address'

FROM Staff JOIN StaffAddress

ON Staff.Ssn = StaffAddress.ssn

WHERE Lname LIKE 'G%';

A close up of a logo

Description generated with very high confidence

1. Retrieves the water monthly cost of all the branches that are more than 2150.

SELECT B\_number AS 'Branch Number', Location, ExpID, monthlyCost

FROM Branch JOIN Has

ON branch.B\_number = has.Bnumber

GROUP BY B\_number, Location, monthlyCost, ExpID

HAVING MonthlyCost > 2150

AND ExpID ='1';

**A close up of a logo

Description generated with very high confidence**

1. Retrieve the MonthlyPayDay for each of the Expenses that belong to Branch 4.

SELECT ExpID, MonthlyPayDay

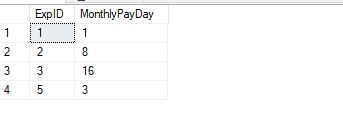
FROM Expenses

WHERE ExpID IN (SELECT Has.ExpID

FROM Has,Branch

WHERE B\_number = Has.Bnumber

AND Bnumber = '4')



**Bonus Trigger Bonus by Omar Atef**

USE Group30

GO

CREATE TRIGGER tr\_Member\_Insert

ON Member

AFTER INSERT,UPDATE,DELETE

AS

BEGIN

PRINT 'A new Member joined the Gym'

END

GO

USE Group30

GO

INSERT INTO Member

VALUES('193255','Platinum','2/3/2018','27/12/2018','Ahmed','Rabee','16/7/1999');

