# Queries(T6):

**SELECT (mID), fNames, sName, mStarted, mExpire**

**FROM Member**

**GROUP BY mStarted**

**HAVING COUNT (mStarted) >= 2017-02-20**

**ORDER BY COUNT (mStarted) ASC**

**- This query lists the members by membership start date and will only show memberships started after 20th February 2017 and will be sorted in ascending order.**

**SELECT AVG (salary), mID, position**

**FROM Staff, Paygrade**

**WHERE position = 'Manager'**

**GROUP BY mID - This query will return the average salary for those employees who have the title manager. This is then grouped by membership ID.**

**RESULT:**

****

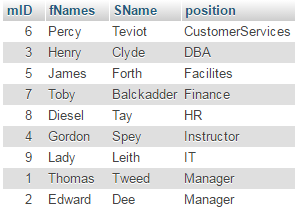
**SELECT Member.mID, Member.fNames, Member.SName, Staff.position**

**FROM Member**

**INNER JOIN Staff**

**ON Member.mID=Staff.mID**

* **This query uses an INNER JOIN to return the staff names, ID’s and there positions.**



**SELECT R.rNumber**

**FROM M.Member,**

**B.Booking,**

**R.Room**

**WHERE M.mID=B.bID**

**AND R.rNumber=B.bID**

**AND M.sName= ’Forth’**

**AND M.fNames = ‘James’**

* **This query searches for bookings made by James Forth.**

# Views(T6):

1. CREATE VIEW MEMBER\_VIEW AS SELECT fNames, sName, mID, dob FROM M ember

* SELECT \* FROM MEMBER\_VIEW - MEMBER\_VIEW was created to create a view from the member table which will be used to show key member information.

1. CREATE VIEW STAFF\_VIEW AS SELECT fNames, sName, mID, position FROM Member, Staff

* SELECT \* FROM STAFF\_VIEW - STAFF\_VIEW was created to create a view from the member and sraff table which will be used to show key staff information.

# Indexes(T7):

1. CREATE INDEX MEMBER\_NAME\_IDX

ON Member (fNames, sName) - The index IDX\_MEMBER\_NAME was created to allow for more efficient queries regarding the member table. Over time the database will increase in size and ultimatley this index will increase in efficiency. This index will have an added benefit as the member table will be consistently updated as the gym grows which is where indexes can help reduce query run time.

1. CREATE INDEX MEMBERID\_NAME\_IDX

ON Member (mID, fNames, sName) - The index IDX\_MEMBERID\_NAME is largely the same as MEMBER\_NAME with the added benefit of having the member ID. This particular index can be useful when using queries that require not only the name but the ID. This can be used in such instances regarding the booking and staff tables.

1. CREATE INDEX STAFF\_POSITION\_IDX

ON Staff (position)

- The index IDX\_STAFF\_POSITION will come into particular use around the staff and paygrade tables. It like the member index will increase in both usefulness and efficiency as the gym staff team grows but as of now the indexes are limited in efficiency due to the small tam size at the gym.

1. CREATE UNIQUE INDEX BOOKING\_SYSTEM\_IDX

ON Booking (bID)

- The unique index IDX\_BOOKING\_SYSTEM is a very useful index to have and differes to standard indexes. This ensures each booking ID is unique and that there annot be a duplicate which could be disasterous for a gym.