ASSIGNMENT 1.1

Qn.1. Get everything in the Sailors table.

* SELECT \* FROM SAILORS;
* A screenshot of a computer

  Description automatically generated

Qn.2. Get Sailor ID, rank and age of all sailors, ordered from highest to lowest rank. Rank is 10 times rating.

* SELECT SID, RATING\*10, AGE FROM SAILORS ORDER BY RATING DESC;
* A screenshot of a computer

  Description automatically generated

Qn.3. Get alphabetical list of sailors with ratings less than 10.

* SELECT Sname from SAILORS WHERE RATING <= 9 ORDER BY Sname;
* A screenshot of a computer

  Description automatically generated

Qn.4. Find how much deposit money is there in total and how many tuples are in the reserves tables.

* SELECT SUM(DEPOSIT) AS TOTAL, COUNT(DEPOSIT) AS HOWMANY FROM RESERVES;
* A screenshot of a computer

  Description automatically generated

Qn.5. Get info on boats in Fishhoek.

* SELECT \* FROM BOATS WHERE LOCATION LIKE “\_IS%K”;
* A screenshot of a computer

  Description automatically generated

Qn.6. In what locations are boats kept?

* SELECT DISTINCT LOCATION FROM BOATS;
* A screenshot of a computer

  Description automatically generated

Qn.7. Get the names of all boats that have fee value recorded in the database.

* SELECT BNAME FROM BOATS WHERE FEE IS NOT NULL;
* A screenshot of a computer

  Description automatically generated

Qn.8. GET ID of all boats that have not been reserved.

* SELECT BID FROM BOATS EXCEPT (SELECT BID FROM RESERVES);
* A screenshot of a computer

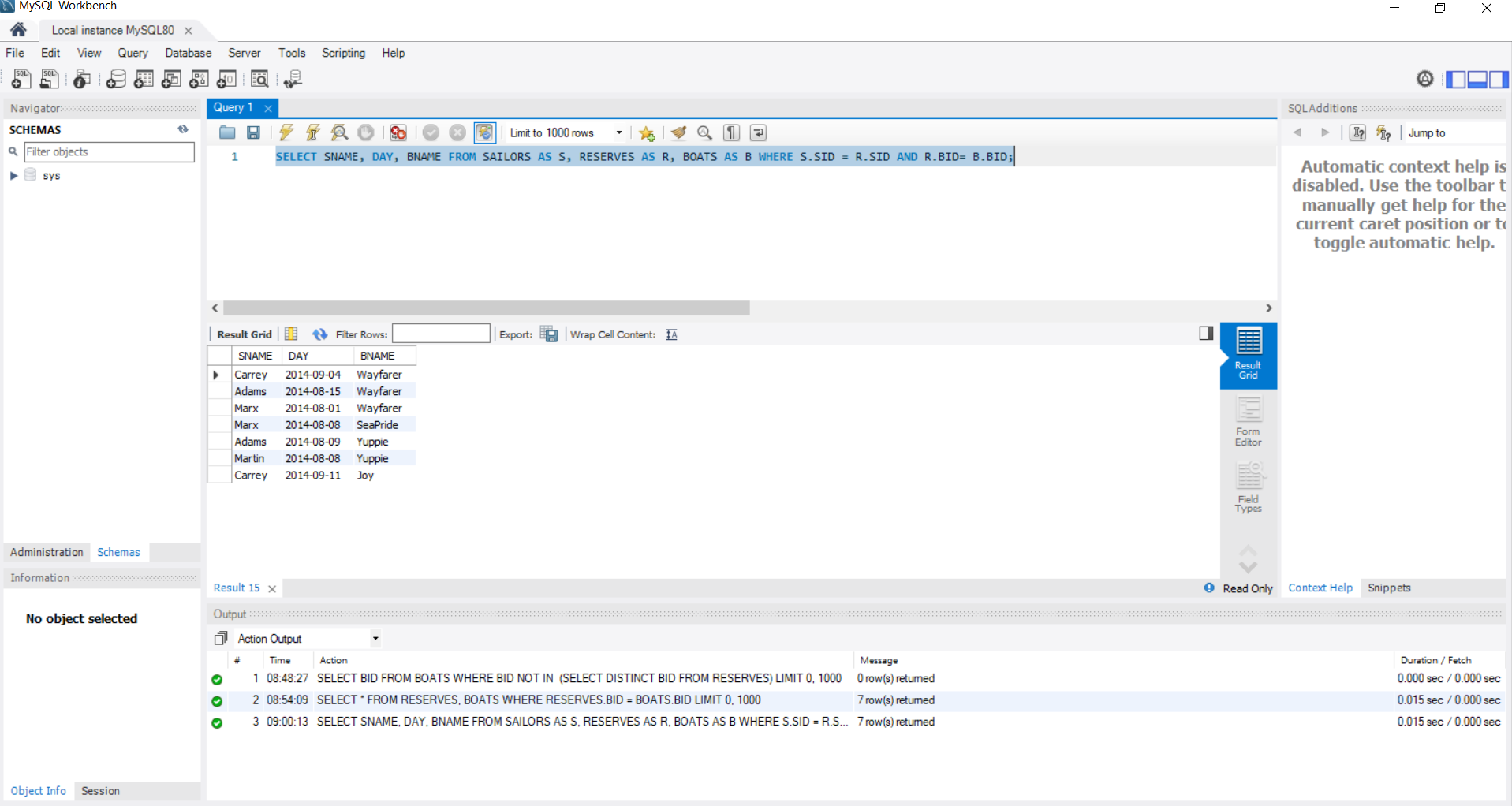
  Description automatically generated

Qn.9. Get all reservation info, including all details on the boats being reserved.

* SELECT \* FROM RESERVES, BOATS WHERE RESERVES.BID = BOATS.BID;
* A screenshot of a computer

  Description automatically generated

Qn.10. For all reservations, get the name of the sailor, along with the day and name of the boat booked.

* SELECT SNAME, DAY, BNAME FROM SAILORS AS S, RESERVES AS R, BOATS AS B WHERE S.SID = R.SID AND R.BID= B.BID;
* 

Qn.11. Get the average deposit paid for each boat.

* SELECT BID, AVG(DEPOSIT) FROM RESERVES GROUP BY BID;
* A screenshot of a computer

  Description automatically generated

Qn.12. Get the average deposit paid for each boat that has been booked by more than one person.

* SELECT BID, AVG(DEPOSIT) FROM RESERVES GROUP BY BID HAVING COUNT(DISTINCT SID)>1;
* A screenshot of a computer

  Description automatically generated

Qn.13. Get the average firm deposit paid for each boat that has been booked by more than one person, in increasing order of amount. A firm deposit is one which exceeds R10.

* SELECT BID, AVG(DEPOSIT) AS AVERAGEDEPOSIT FROM RESERVES WHERE DEPOSIT>10 GROUP BY BID HAVING COUNT (DISTINCT SID)>1 ORDER BY AVERAGEDEPOIST;
* A screenshot of a computer

  Description automatically generated

Qn.14. Get name and rating of sailors with rating exceeding 7 who made any reservation with 0 deposit.

* SELECT SNAME, RATING FROM SAILORS WHERE RATING>7 AND SID IN (SELECT SID FROM RESERVES WHERE DEPOSIT =0 );
* A screenshot of a computer

  Description automatically generated

Qn.15. Get names of boats located in a place other than Hout Bay or Fish Hoek.

* SELECT BNAME FROM BOATS WHERE LOCATION NOT IN (‘HOUT BAY’, ‘FISH HOEK’);
* A screenshot of a computer

  Description automatically generated

Qn.16. Get names of boats having a fee larger than any boat located in Hout Bay.

* SELECT DISTINCT BNAME FROM BOATS WHERE FEE> SOME (SELECT FEE FROM BOATS WHERE LOCATION = “HOUT BAY”);
* A screenshot of a computer

  Description automatically generated

Qn.17. Get the names that are in both the sailors and captains relations.

* SELECT SNAME FROM SAILORS WHERE EXISTS (SELECT \* FROM CAPTAINS WHERE CAPTAINS.SID=SAILORS.SID);
* A screenshot of a computer

  Description automatically generated

Qn.18. Get names of boats that have exactly one reservation.

* SELECT BNAME FROM BOATS AS B WHERE B.BID IN (SELECT BID FROM RESERVES GROUP BY BID HAVING COUNT(BID)=1);A screenshot of a computer

  Description automatically generated

Qn.19. Get Sailor ID and total deposit paid for sailors who have booked more than 1 boat.

* SELECT SID, TOTAL\_DEPOSIT FROM (SELECT SID, COUNT(BID), SUM(DEPOSIT) FROM RESERVES WHERE DEPOSIT IS NOT NULL AND DEPOSIT > 0 GROUP BY SID ) AS RESULT(SID, NUM\_BOATS, TOTAL\_DEPOSIT) WHERE NUM\_BOATS > 1;
* A screenshot of a computer

  Description automatically generated

QN.20. Get all reservation info including details of the boat booked.

* SELECT \* FROM BOATS INNER JOIN RESERVES ON BOATS.BID= RESERVES.BID;
* A screenshot of a computer

  Description automatically generated

Qn.21. Get all information on every boat. If a boat has reservations, including all its reservation info.

* SELECT \* FROM BOATS LEFT OUTER JOIN RESERVES ON BOATS.BID=RESERVES.BID;
* A screenshot of a computer

  Description automatically generated

Qn.22. Create a new Tuple for the boat named “Nino” which has fee R150, BID 110, and is in Fish Hoek.

* INSERT INTO BOATS VALUES ("Nino", 110, 150, "Fish Hoek");
* A screenshot of a computer

  Description automatically generated

Qn.23. Delete from reserves where deposit is Null or Deposit =0.

* DELETE FROM RESERVES WHERE DEPOSIT IS NULL OR DEPOSIT = 0;
* A screenshot of a computer

  Description automatically generated

Qn.24. Increase the fee of every boat by 12%.

* UPDATE BOATS SET FEE= FEE\*1.12;
* A screenshot of a computer

  Description automatically generated

Qn.25. Make a view called Bookings which hides the Deposit value i.e., only has the other 3 attributes.

* CREATE VIEW BOOKINGS AS SELECT SID, BID, DAY FROM RESERVES;
* A screenshot of a computer

  Description automatically generated

Qn.26. Create a table called reserves with 3 integer values BID, SID & Deposit, and a date attribute Day. Allow only deposit to be omitted, and ensure SID and BID values exist in the database. When someone books a boat it is for the whole day.

* CREATE TABLE RESERVES (BID INTEGER NOT NULL, SID INTEGER NOT NULL, DAY DATETIME NOT NULL, DEPOSIT INTEGER, PRIMARY KEY (BID, SID), CHECK (BID IN (SELECT BID FROM BOATS)), CHECK (SID IN (SELECT SID FROM SAILORS) ));
* A screenshot of a computer

  Description automatically generated

Qn.27. Add a new attribute NEEDSREPAIR to the boats table. It is usually “N”

* ALTER TABLE BOATS ADD NEEDSREPAIR CHAR(1) DEFAULT “N”;
* A screenshot of a computer

  Description automatically generated

Qn.28. We should not be ageists. Please remove age attribute.

* ALTER TABLE SAILORS DROP AGE;
* A screenshot of a computer

  Description automatically generated

Qn.29. Remove the captains relation altogether so that no one tries to update or insert captain in the future.

* DROP TABLE CAPTAINS;
* A screenshot of a computer

  Description automatically generated

# **Part 2 – Relation Algebra**

Qn.2. Get Sailor ID, rank and age of all sailors, ordered from highest to lowest rank. Rank is 10 times rating.

* SELECT SID, RATING\*10, AGE FROM SAILORS ORDER BY RATING DESC;
* πSID, RATING\*10 AS RANK, AGE ,RANK DESC

Qn.8. GET ID of all boats that have not been reserved.

* SELECT BID FROM BOATS EXCEPT (SELECT BID FROM RESERVES);
* πBID (BOATS)-πBID(σBID(RESERVES))

Qn.9. Get all reservation info, including all details on the boats being reserved.

* SELECT \* FROM RESERVES, BOATS WHERE RESERVES.BID = BOATS.BID;
* RESERVES ⨝RESERVES.BID=BOATS.BID BOATS

Qn.10. For all reservations, get the name of the sailor, along with the day and name of the boat booked.

* SELECT SNAME, DAY, BNAME FROM SAILORS AS S, RESERVES AS R, BOATS AS B WHERE S.SID = R.SID AND R.BID= B.BID;
* πSNAME,DAY,BNAME(σSAILORS.SID=RESERVES.SID Λ RESERVES.BID=BOATS.BID)(SAILORS⨝ RESERVES⨝ BOATS)

Qn.17. Get the names that are in both the sailors and captains relations.

* SELECT SNAME FROM SAILORS WHERE EXISTS (SELECT \* FROM CAPTAINS WHERE CAPTAINS.SID=SAILORS.SID);
* πSNAME(σSAILORS.SID=CAPTAINS.SID)(SAILORS ⨝ CAPTAINS)

Qn.20. Get all reservation info including details of the boat booked.

* SELECT \* FROM BOATS INNER JOIN RESERVES ON BOATS.BID= RESERVES.BID;
* σBOATS.BID=RESERVES.BID (BOATS ⨝ RESERVES)