SQL STATEMENTS

CREATING SCHEMA

Database

```
CREATE DATABASE dartdb;
```

CREATING TABLES

AClass_Pass

```
CREATE TABLE `aclass_pass` (
  `passid` int NOT NULL,
  `pid` varchar(4) NOT NULL,
  `month` varchar(10) NOT NULL,
  `year` varchar(10) NOT NULL,
  PRIMARY KEY (`month`,`pid`,`year`),
  KEY `passid` (`passid`),
  KEY `pid` (`pid`),
  CONSTRAINT `aclass_pass_ibfk_1` FOREIGN KEY (`passid`) REFERENCES `pass` (`passid`),
  CONSTRAINT `aclass_pass_ibfk_2` FOREIGN KEY (`pid`) REFERENCES `aclass_passenger` (`pid`)
)
```

AClass_Passenger

```
CREATE TABLE `aclass_passenger` (
  `pid` varchar(4) NOT NULL,
  PRIMARY KEY (`pid`),
  CONSTRAINT `aclass_passenger_ibfk_1` FOREIGN KEY (`pid`) REFERENCES `person` (`pid`)
)
```

AStar_Pass

```
CREATE TABLE `astar_pass` (
  `passid` int NOT NULL,
  `pid` varchar(4) NOT NULL,
  PRIMARY KEY (`passid`),
  KEY `pid` (`pid`),
  CONSTRAINT `astar_pass_ibfk_1` FOREIGN KEY (`passid`) REFERENCES `pass` (`passid`),
  CONSTRAINT `astar_pass_ibfk_2` FOREIGN KEY (`pid`) REFERENCES `astar_passenger` (`pid`)
)
```

AStar_Passenger

```
CREATE TABLE `astar_passenger` (
  `pid` varchar(4) NOT NULL,
  PRIMARY KEY (`pid`),
  CONSTRAINT `astar_passenger_ibfk_1` FOREIGN KEY (`pid`) REFERENCES `person` (`pid`)
)
```

Bus

```
CREATE TABLE `bus` (
  `busno` int NOT NULL,
  `licno` varchar(20) NOT NULL,
  `noofseats` int NOT NULL,
  `ttid` varchar(4) NOT NULL,
  `routeid` int NOT NULL,
  `eid` varchar(4) NOT NULL,
  PRIMARY KEY (`busno`),
  KEY `ttid` (`ttid`),
  KEY `routeid` (`routeid`),
  KEY `eid` (`eid`),
```

```
  CONSTRAINT `bus_ibfk_1` FOREIGN KEY (`ttid`) REFERENCES `timetable` (`ttid`),
  CONSTRAINT `bus_ibfk_2` FOREIGN KEY (`routeid`) REFERENCES `route` (`routeid`),
  CONSTRAINT `bus_ibfk_3` FOREIGN KEY (`eid`) REFERENCES `employee` (`eid`)
)
```

BusStop

```
CREATE TABLE `busstop` (
  `routeid` int NOT NULL,
  `busstopid` int NOT NULL,
  PRIMARY KEY (`routeid`,`busstopid`),
  CONSTRAINT `busstop_ibfk_1` FOREIGN KEY (`routeid`) REFERENCES `route` (`routeid`)
)
```

Drives

```
CREATE TABLE `drives` (
  `eid` varchar(4) NOT NULL,
  `busno` int NOT NULL,
  `date` date NOT NULL,
  PRIMARY KEY (`busno`,`eid`,`date`),
  KEY `eid` (`eid`),
  CONSTRAINT `drives_ibfk_1` FOREIGN KEY (`busno`) REFERENCES `bus` (`busno`),
  CONSTRAINT `drives_ibfk_2` FOREIGN KEY (`eid`) REFERENCES `employee` (`eid`)
)
```

Employee

```
CREATE TABLE `employee` (
  `eid` varchar(4) NOT NULL,
  `startdate` date NOT NULL,
  `type` varchar(20) NOT NULL,
```

```
  PRIMARY KEY (`eid`),
  CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`eid`) REFERENCES `person` (`pid`),
  CONSTRAINT `employee_chk_1` CHECK ((`type` in (’Staff', 'Ticket Checker', 'Driver')))
)
```

Guest

```
CREATE TABLE `guest` (
  `guestSSN` varchar(15) NOT NULL,
  `gname` varchar(30) NOT NULL,
  `gaddress` varchar(120) NOT NULL,
  `gcontact` int NOT NULL,
  PRIMARY KEY (`guestSSN`)
)
```

GuestLog

```
CREATE TABLE `guestlog` (
  `pid` varchar(4) NOT NULL,
  `month` varchar(20) NOT NULL,
  `year` int NOT NULL,
  `count` int NOT NULL,
  `guestid` varchar(20) NOT NULL,
  `guestSSN` varchar(15) NOT NULL,
  PRIMARY KEY (`pid`,`month`,`year`),
  KEY `guestSSN` (`guestSSN`),
  CONSTRAINT `guestlog_ibfk_1` FOREIGN KEY (`pid`) REFERENCES `astar_passenger` (`pid`),
  CONSTRAINT `guestlog_ibfk_2` FOREIGN KEY (`guestSSN`) REFERENCES `guest` (`guestSSN`),
  CONSTRAINT `guestlog_chk_1` CHECK ((`count` between 1 and 4))
)
```

Pass

```
CREATE TABLE `pass` (
  `passid` int NOT NULL,
  `validity` date NOT NULL,
  `eid` varchar(4) NOT NULL,
  PRIMARY KEY (`passid`),
  KEY `eid` (`eid`),
  CONSTRAINT `pass_ibfk_1` FOREIGN KEY (`eid`) REFERENCES `employee` (`eid`)
)
```

Payment

```
CREATE TABLE `payment` (
  `paymentid` int NOT NULL AUTO_INCREMENT,
  `method` varchar(120) NOT NULL,
  `amount` float NOT NULL,
  PRIMARY KEY (`paymentid`),
  CONSTRAINT `payment_chk_1` CHECK ((`method` in ('Cash', 'Card')))
)
```

Person

```
CREATE TABLE `person` (
  `pid` varchar(4) NOT NULL,
  `fname` varchar(120) NOT NULL,
  `mname` varchar(120) DEFAULT NULL,
  `lname` varchar(120) NOT NULL,
  `address` varchar(300) NOT NULL,
  `gender` varchar(10) NOT NULL,
  `dob` date NOT NULL,
  PRIMARY KEY (`pid`),
```

```
  CONSTRAINT `GCHECK` CHECK ((`gender` in ('M', 'F'))),
  CONSTRAINT `PIDCHECK` CHECK (regexp_like(`pid`,'^[P][0-9]{3}$'))
)
```

PersonPhoneNumber

```
CREATE TABLE `personphonenumber` (
  `pid` varchar(4) NOT NULL,
  `phonenumber` int NOT NULL,
  PRIMARY KEY (`pid`,`phonenumber`),
  CONSTRAINT `personphonenumber_ibfk_1` FOREIGN KEY (`pid`) REFERENCES `person` (`pid`),
  CONSTRAINT `personphonenumber_chk_1` CHECK ((`phonenumber` between 100000000 and 9999999999))
)
```

Promotion

```
CREATE TABLE `promotion` (
  `promoid` int NOT NULL,
  `cardid` int NOT NULL,
  `description` varchar(60) DEFAULT NULL,
  `validity` date NOT NULL,
  PRIMARY KEY (`promoid`,`cardid`),
  KEY `cardid` (`cardid`),
  CONSTRAINT `promotion_ibfk_1` FOREIGN KEY (`cardid`) REFERENCES `travelcard` (`cardid`)
)
```

Route

```
CREATE TABLE `route` (
  `routeid` int NOT NULL AUTO_INCREMENT,
  `routename` varchar(120) NOT NULL,
```

```
  PRIMARY KEY (`routeid`)
)
```

Terminal

```
CREATE TABLE `terminal` (
  `terminalid` int NOT NULL AUTO_INCREMENT,
  `location` varchar(300) NOT NULL,
  `date` date NOT NULL,
  `time` time NOT NULL,
  PRIMARY KEY (`terminalid`)
)
```

Ticket

```
CREATE TABLE `ticket` (
  `ticketid` int NOT NULL AUTO_INCREMENT,
  `busno` int NOT NULL,
  `seatno` int NOT NULL,
  `price` int NOT NULL,
  PRIMARY KEY (`ticketid`),
  KEY `busno_idx` (`busno`),
  CONSTRAINT `busno` FOREIGN KEY (`busno`) REFERENCES `bus` (`busno`)
)
```

TicketSales

```
CREATE TABLE `ticketsales` (
  `paymentid` int NOT NULL,
  `ticketid` int NOT NULL,
  `eid` varchar(4) NOT NULL,
  `pid` varchar(4) NOT NULL,
```

```
  `bookingdate` date NOT NULL,
  PRIMARY KEY (`paymentid`,`ticketid`,`eid`,`pid`),
  KEY `ticketid` (`ticketid`),
  KEY `eid` (`eid`),
  KEY `pid` (`pid`),
  CONSTRAINT `ticketsales_ibfk_1` FOREIGN KEY (`ticketid`) REFERENCES `ticket` (`ticketid`),
  CONSTRAINT `ticketsales_ibfk_2` FOREIGN KEY (`paymentid`) REFERENCES `payment` (`paymentid`),
  CONSTRAINT `ticketsales_ibfk_3` FOREIGN KEY (`eid`) REFERENCES `employee` (`eid`),
  CONSTRAINT `ticketsales_ibfk_4` FOREIGN KEY (`pid`) REFERENCES `person` (`pid`)
)
```

TimeTable

```
CREATE TABLE `timetable` (
  `ttid` varchar(4) NOT NULL,
  `day` varchar(10) NOT NULL,
  `starttime` time NOT NULL,
  `endtime` time NOT NULL,
  `interval` int NOT NULL,
  PRIMARY KEY (`ttid`),
  CONSTRAINT `timetable_chk_1` CHECK (regexp_like(`ttid`,'^(DT)?[0-9]{2}$')),
  CONSTRAINT `timetable_chk_2` CHECK ((`day` in ('M','T','W','Th','F','Sat', 'Sun'))),
  CONSTRAINT `timetable_chk_3` CHECK ((`interval` in (15,20,30)))
)
```

TravelCard

```
CREATE TABLE `travelcard` (
  `cardid` int NOT NULL AUTO_INCREMENT,
  `pid` varchar(4) NOT NULL,
  `date_of_issue` date NOT NULL,
```

```
  `validity` date NOT NULL,
  PRIMARY KEY (`cardid`,`pid`),
  KEY `travelcard_ibfk_1` (`pid`),
  CONSTRAINT `travelcard_ibfk_1` FOREIGN KEY (`pid`) REFERENCES `astar_passenger` (`pid`)
)
```

CREATING TRIGGERS

Person

```
CREATE DEFINER=`root`@`localhost` TRIGGER `person_BEFORE_INSERT` BEFORE INSERT ON `person` FOR EACH ROW BEGIN
IF TIMESTAMPDIFF(YEAR, NEW.dob,CURDATE()) < 16 THEN
 SIGNAL SQLSTATE '45000'
 SET MESSAGE_TEXT = 'AGE MUST BE GREATER THAN 16';
END IF;
END
```

QUERIES

1. For each employee class, list the employees belonging to that class.

```
SELECT e.type, e.eid, p.fname, p.lname, p.gender
FROM employee e, person p
WHERE p.pid = e.eid
ORDER BY e.type;
```

2. Find the names of employees who are also an A-Class Passenger.

```
SELECT p.fname, p.lname
FROM employee e, person p, aclass_passenger a
WHERE e.eid = a.pid AND e.eid = p.pid;
```

3. Find the average number of bookings made by the top five A-Star Passengers.

```
SELECT AVG(bookings)
FROM ( SELECT *
        FROM top_astar_passenger
        ORDER BY bookings DESC
        LIMIT 5 ) as TOPFIVE;
```

4. Find the Bus ID and Route names of the bus that is booked the most.

```
SELECT b.busno, b.routeid
FROM ticketsales ts, ticket t, bus b
WHERE ts.ticketid = t.ticketid AND t.busno = b.busno
GROUP BY  b.busno, b.routeid
ORDER BY COUNT(*) DESC
LIMIT 1;
```

6. Find the total number bookings for each bus in the system.

```
SELECT b.busno, count(ts.pid) AS Bookings
FROM ( ticketsales ts INNER JOIN ticket t ON ts.ticketid = t.ticketid ) RIGHT JOIN bus b ON t.busno = b.busno
GROUP BY  b.busno
ORDER BY COUNT(ts.pid);
```

7. Find the driver details who has driven every day of the past week.

```
SELECT d.eid
FROM DRIVES d, timetable t, bus b
WHERE d.busno = b.busno
        AND b.ttid = t.ttid
        AND d.date BETWEEN (CURRENT_DATE - INTERVAL 7 DAY) AND CURRENT_DATE
        AND t.day IN ( SELECT DISTINCT(day) FROM timetable )
GROUP BY d.eid
HAVING COUNT(d.eid) =  ( SELECT COUNT(DISTINCT(day)) FROM timetable )
```

8. Find the count of passengers who booked the most popular bus.

```
SELECT COUNT(DISTINCT(s.pid)) AS BOOKINGS
FROM popular_bus pb, ticket t, ticketsales s
WHERE s.ticketid = t.ticketid AND t.busno = pb.busno ;
```

9. List all the booking details issued after the most current employee was hired.

```
SELECT *
FROM ticketsales
WHERE bookingdate > ( SELECT startdate FROM employee ORDER BY startdate DESC LIMIT 1 );
```

10. List all the employees that have enrolled as A-Star Passengers within a month of being employed.

```
SELECT e.eid, p.fname, e.startdate, tc.date_of_issue, tc.cardid
FROM employee e,
astar_passenger asp, travelcard tc, person p
WHERE e.eid = p.pid
        AND e.eid = asp.pid
        AND tc.pid = asp.pid
        AND tc.date_of_issue BETWEEN e.startdate AND DATE(e.startdate + INTERVAL 1 MONTH);
```

11. Find the route with the highest number of bus stops.

```
SELECT routeid, count(*) AS BusStopCount
FROM busstop
 GROUP BY routeid
ORDER BY BusStopCount DESC
LIMIT 1;
```

12. Find the name of passengers who have been A-Star Passengers for over 5 years.

```
SELECT p.fname, tc.date_of_issue
FROM person p, astar_passenger asp, travelcard tc
WHERE p.pid = asp.pid
        AND asp.pid = tc.pid
        AND tc.date_of_issue < DATE(CURRENT_DATE - INTERVAL 5 YEAR);
```

13. Find the bookings made by the potential A-Star Passengers in the last year.

```
SELECT pap.pid, COUNT(*) AS Bookings
FROM potential_astar_passenger pap, ticketsales ts
WHERE pap.pid = ts.pid
        AND ts.bookingdate BETWEEN (CURRENT_DATE - INTERVAL 1 YEAR) AND CURRENT_DATE
GROUP BY pap.pid ;
```

VIEWS

1. Top A-Star Passenger- This view returns the First Name, Last Name and Date of membership enrollment of those passengers who have travelled more than 6 times in the last month.

```
CREATE VIEW `top_astar_passenger` AS
   SELECT
      `p`.`pid` AS `pid`,
      `p`.`fname` AS `fname`,
      `p`.`lname` AS `lname`,
      `tc`.`date_of_issue` AS `date_of_issue`,
      COUNT(0) AS `bookings`
   FROM
      ((`person` `p`
      JOIN `travelcard` `tc`)
      JOIN `ticketsales` `ts`)
   WHERE
      ((`p`.`pid` = `tc`.`pid`)
         AND (`ts`.`pid` = `tc`.`pid`)
         AND (`ts`.`bookingdate` BETWEEN CAST((CURDATE() - INTERVAL 1 MONTH) AS DATE) AND
CURDATE()))
   GROUP BY `p`.`pid` , `p`.`fname` , `p`.`lname` , `tc`.`date_of_issue`
   HAVING (COUNT(*) > 6)
```

2. Popular Bus- This view returns the details of the bus that the passenger has booked the most in the past 2 months

```
CREATE  VIEW `popular_bus` AS
   SELECT
      `b`.`busno` AS `busno`,
      `b`.`licno` AS `licno`,
      `b`.`noofseats` AS `noofseats`,
      `b`.`ttid` AS `ttid`,
      `b`.`routeid` AS `routeid`,
      `b`.`eid` AS `eid`
   FROM
      ((`ticketsales` `ts`
      JOIN `ticket` `t`)
      JOIN `bus` `b`)
   WHERE
      ((`ts`.`ticketid` = `t`.`ticketid`)
         AND (`t`.`busno` = `b`.`busno`)
         AND (`ts`.`bookingdate` BETWEEN (CURDATE() - INTERVAL 2 MONTH) AND CURDATE()))
   GROUP BY `b`.`busno` , `b`.`licno` , `b`.`noofseats` , `b`.`ttid` , `b`.`routeid` , `b`.`eid`
   ORDER BY COUNT(*) DESC
   LIMIT 1
```

4. Potential A-Star Passenger- This view returns the name, phone number and ID of the A-Class Passengers who travelled more than 4 time in the past 2 months.

```
CREATE VIEW `potential_astar_passenger` AS
  SELECT
    `p`.`fname` AS `fname`,
    `ppn`.`phonenumber` AS `phonenumber`,
    `p`.`pid` AS `pid`
  FROM
    (((`person` `p`
    JOIN `personphonenumber` `ppn`)
    JOIN `aclass_passenger` `acp`)
    JOIN `ticketsales` `ts`)
  WHERE
    ((`p`.`pid` = `ppn`.`pid`)
      AND (`p`.`pid` = `acp`.`pid`)
      AND (`acp`.`pid` = `ts`.`pid`)
      AND (`ts`.`bookingdate` BETWEEN (CURDATE() - INTERVAL 2 MONTH) AND CURDATE()))
  GROUP BY `p`.`fname` , `ppn`.`phonenumber` , `p`.`pid`
  HAVING (COUNT(*) > 4)
```

5. Top Employee- This view returns the details of the employee who has made the most number of bookings in the past month.

```
CREATE VIEW `top_employee` AS
   SELECT
      `p`.`pid` AS `pid`,
      `p`.`fname` AS `fname`,
      `p`.`mname` AS `mname`,
      `p`.`lname` AS `lname`,
      `p`.`address` AS `address`,
      `p`.`gender` AS `gender`
   FROM
      ((`person` `p`
      JOIN `employee` `e`)
      JOIN `ticketsales` `ts`)
   WHERE
      ((`p`.`pid` = `e`.`eid`)
         AND (`e`.`eid` = `ts`.`eid`)
         AND (`ts`.`bookingdate` BETWEEN (CURDATE() - INTERVAL 1 MONTH) AND CURDATE()))
   GROUP BY `p`.`fname` , `p`.`pid`
   ORDER BY COUNT(*) DESC
   LIMIT 1
```