**f. SQL Statements (Solution of Phase 3 – c d e)**

|  |  |  |
| --- | --- | --- |
| **CREATE SCHEMA dart;** | | |
| **Table Creation** | | |
| **No.** | **Table** | **SQL** |
|  | | |
| **1** | **person** | CREATE TABLE `person` (  `person\_id` varchar(4) NOT NULL,  `f\_name` varchar(45) NOT NULL,  `m\_name` varchar(45) DEFAULT NULL,  `l\_name` varchar(45) NOT NULL,  `gender` varchar(1) NOT NULL,  `dob` date NOT NULL,  `street` varchar(45) NOT NULL,  `apt\_no` varchar(5) NOT NULL,  `zip\_code` varchar(5) NOT NULL,  PRIMARY KEY (`person\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **2** | **person\_**  **phone** | CREATE TABLE `person\_phone` (  `phn\_person\_id` varchar(4) NOT NULL,  `phone\_no` varchar(10) NOT NULL,  PRIMARY KEY (`phn\_person\_id`,`phone\_no`),  CONSTRAINT `fk\_person\_phone\_1` FOREIGN KEY (`phn\_person\_id`) REFERENCES `person` (`person\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **3** | **zip\_code** | CREATE TABLE `dart`.`zip\_code` (  `zip\_code` VARCHAR(5) NOT NULL,  `city` VARCHAR(45) NOT NULL,  PRIMARY KEY (`zip\_code`)); |
|  | | |
| **4** | **a\_star\_**  **passenger** | CREATE TABLE `a\_star\_passenger` (  `a\_star\_id` varchar(5) NOT NULL,  PRIMARY KEY (`a\_star\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **5** | **employee** | CREATE TABLE `employee` (  `employee\_id` varchar(5) NOT NULL,  `emp\_person\_id` varchar(5) NOT NULL,  `start\_date` date NOT NULL,  `e\_type` varchar(45) NOT NULL,  PRIMARY KEY (`employee\_id`,`emp\_person\_id`),  KEY `emp\_person\_id` (`emp\_person\_id`),  CONSTRAINT `employee\_ibfk\_1` FOREIGN KEY (`emp\_person\_id`) REFERENCES `person` (`person\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **6** | **a\_class\_**  **passenger** | CREATE TABLE `a\_class\_passenger` (  `passenger\_id` varchar(5) NOT NULL,  `ac\_person\_id` varchar(5) NOT NULL,  PRIMARY KEY (`passenger\_id`,`ac\_person\_id`),  KEY `fk\_a\_class\_passenger\_1\_idx` (`ac\_person\_id`),  CONSTRAINT `fk\_a\_class\_passenger\_1` FOREIGN KEY (`ac\_person\_id`) REFERENCES `person` (`person\_id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **7** | **staff** | CREATE TABLE `staff` (  `staff\_id` varchar(5) NOT NULL,  `s\_employee\_id` varchar(5) NOT NULL,  PRIMARY KEY (`staff\_id`,`s\_employee\_id`),  KEY `fk\_staff\_1\_idx` (`s\_employee\_id`),  CONSTRAINT `fk\_staff\_1` FOREIGN KEY (`s\_employee\_id`) REFERENCES `employee` (`employee\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **8** | **ticket\_**  **checker** | CREATE TABLE `ticket\_checker` (  `checker\_id` varchar(5) NOT NULL,  `tc\_bus\_no` varchar(7) NOT NULL,  `tc\_employee\_id` varchar(5) NOT NULL,  PRIMARY KEY (`checker\_id`,`tc\_bus\_no`,`tc\_employee\_id`),  KEY `fk\_ticket\_checker\_1\_idx` (`tc\_employee\_id`),  KEY `fk\_ticket\_checker\_2\_idx` (`tc\_bus\_no`),  CONSTRAINT `fk\_ticket\_checker\_1` FOREIGN KEY (`tc\_employee\_id`) REFERENCES `employee` (`employee\_id`),  CONSTRAINT `fk\_ticket\_checker\_2` FOREIGN KEY (`tc\_bus\_no`) REFERENCES `bus` (`license\_plate\_no`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **9** | **bus\_driver** | CREATE TABLE `bus\_driver` (  `driver\_id` varchar(5) NOT NULL,  `d\_employee\_id` varchar(5) NOT NULL,  PRIMARY KEY (`driver\_id`,`d\_employee\_id`),  KEY `fk\_bus\_driver\_1\_idx` (`d\_employee\_id`),  CONSTRAINT `fk\_bus\_driver\_1` FOREIGN KEY (`d\_employee\_id`) REFERENCES `employee` (`employee\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **10** | **route** | CREATE TABLE `route` (  `route\_id` varchar(5) NOT NULL,  PRIMARY KEY (`route\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;CREATE TABLE `route` (  `route\_id` varchar(5) NOT NULL,  PRIMARY KEY (`route\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **11** | **bus** | CREATE TABLE `bus` (  `license\_plate\_no` varchar(7) NOT NULL,  `bus\_no` varchar(5) NOT NULL,  `bus\_route\_id` varchar(5) NOT NULL,  `no\_of\_seats` int NOT NULL,  `bus\_type` varchar(5) NOT NULL,  PRIMARY KEY (`license\_plate\_no`,`bus\_no`),  KEY `fk\_bus\_2\_idx` (`bus\_route\_id`),  CONSTRAINT `fk\_bus\_2` FOREIGN KEY (`bus\_route\_id`) REFERENCES `route` (`route\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **12** | **ticket** | CREATE TABLE `ticket` (  `ticket\_id` varchar(5) NOT NULL,  `t\_bus\_no` varchar(7) NOT NULL,  `t\_seat\_no` int NOT NULL,  `t\_checker\_id` varchar(5) NOT NULL,  `t\_staff\_id` varchar(5) NOT NULL,  `t\_person\_id` varchar(4) NOT NULL,  `t\_payment\_id` varchar(5) NOT NULL,  `date` date NOT NULL,  PRIMARY KEY (`ticket\_id`),  KEY `fk\_ticket\_1\_idx` (`t\_bus\_no`),  KEY `fk\_ticket\_2\_idx` (`t\_checker\_id`),  KEY `fk\_ticket\_3\_idx` (`t\_staff\_id`),  KEY `fk\_ticket\_4\_idx` (`t\_person\_id`),  KEY `fk\_ticket\_5\_idx` (`t\_payment\_id`),  CONSTRAINT `fk\_ticket\_1` FOREIGN KEY (`t\_bus\_no`) REFERENCES `bus` (`license\_plate\_no`),  CONSTRAINT `fk\_ticket\_2` FOREIGN KEY (`t\_checker\_id`) REFERENCES `ticket\_checker` (`checker\_id`),  CONSTRAINT `fk\_ticket\_3` FOREIGN KEY (`t\_staff\_id`) REFERENCES `staff` (`staff\_id`),  CONSTRAINT `fk\_ticket\_4` FOREIGN KEY (`t\_person\_id`) REFERENCES `person` (`person\_id`),  CONSTRAINT `fk\_ticket\_5` FOREIGN KEY (`t\_payment\_id`) REFERENCES `payment\_details` (`payment\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **13** | **bus\_stop** | CREATE TABLE `bus\_stop` (  `stop\_no` varchar(5) NOT NULL,  `stop\_route\_id` varchar(5) NOT NULL,  `location` varchar(45) NOT NULL,  PRIMARY KEY (`stop\_no`),  KEY `fk\_bus\_stop\_1\_idx` (`stop\_route\_id`),  CONSTRAINT `fk\_bus\_stop\_1` FOREIGN KEY (`stop\_route\_id`) REFERENCES `route` (`route\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **14** | **timetable** | CREATE TABLE `timetable` (  `tt\_id` varchar(4) NOT NULL,  `day` varchar(3) NOT NULL,  `start\_time` time NOT NULL,  `end\_time` time NOT NULL,  `interval` int NOT NULL,  PRIMARY KEY (`tt\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **15** | **payment\_**  **details** | CREATE TABLE `payment\_details` (  `payment\_id` varchar(5) NOT NULL,  `amount` float NOT NULL,  `method` varchar(5) NOT NULL,  PRIMARY KEY (`payment\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **16** | **follows** | CREATE TABLE `follows` (  `f\_bus\_no` varchar(7) NOT NULL,  `f\_tt\_id` varchar(5) NOT NULL,  PRIMARY KEY (`f\_bus\_no`,`f\_tt\_id`),  KEY `fk\_follows\_1\_idx` (`f\_tt\_id`),  CONSTRAINT `fk\_follows\_1` FOREIGN KEY (`f\_tt\_id`) REFERENCES `timetable` (`tt\_id`),  CONSTRAINT `fk\_follows\_2` FOREIGN KEY (`f\_bus\_no`) REFERENCES `bus` (`license\_plate\_no`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **17** | **terminal** | CREATE TABLE `terminal` (  `terminal\_id` varchar(5) NOT NULL,  `location` varchar(45) NOT NULL,  PRIMARY KEY (`terminal\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **18** | **parks** | CREATE TABLE `parks` (  `p\_terminal\_id` varchar(5) NOT NULL,  `p\_bus\_no` varchar(7) NOT NULL,  `time` time NOT NULL,  `date` date NOT NULL,  `duration` int NOT NULL,  PRIMARY KEY (`p\_terminal\_id`,`p\_bus\_no`),  KEY `fk\_parks\_2\_idx` (`p\_bus\_no`),  CONSTRAINT `fk\_parks\_1` FOREIGN KEY (`p\_terminal\_id`) REFERENCES `terminal` (`terminal\_id`),  CONSTRAINT `fk\_parks\_2` FOREIGN KEY (`p\_bus\_no`) REFERENCES `bus` (`license\_plate\_no`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **19** | **pass** | CREATE TABLE `pass` (  `pass\_id` varchar(5) NOT NULL,  `issue\_date` date NOT NULL,  `expiry\_date` date NOT NULL,  `p\_staff\_id` varchar(5) NOT NULL,  `p\_payment\_id` varchar(5) NOT NULL,  PRIMARY KEY (`pass\_id`),  KEY `fk\_pass\_1\_idx` (`p\_staff\_id`),  KEY `fk\_pass\_3\_idx` (`p\_payment\_id`),  CONSTRAINT `fk\_pass\_1` FOREIGN KEY (`p\_staff\_id`) REFERENCES `staff` (`staff\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT,  CONSTRAINT `fk\_pass\_3` FOREIGN KEY (`p\_payment\_id`) REFERENCES `payment\_details` (`payment\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **20** | **Sells\_pass** | CREATE TABLE `sells\_passes` (  `sp\_staff\_id` varchar(5) NOT NULL,  `sp\_pass\_id` varchar(5) NOT NULL,  `date` date NOT NULL,  PRIMARY KEY (`sp\_staff\_id`,`sp\_pass\_id`),  KEY `fk\_sells\_passes\_2\_idx` (`sp\_pass\_id`),  CONSTRAINT `fk\_sells\_passes\_1` FOREIGN KEY (`sp\_staff\_id`) REFERENCES `staff` (`staff\_id`),  CONSTRAINT `fk\_sells\_passes\_2` FOREIGN KEY (`sp\_pass\_id`) REFERENCES `pass` (`pass\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  |  |  |
| **21** | **Sells\_tickets** | CREATE TABLE `sells\_tickets` (  `st\_staff\_id` varchar(5) NOT NULL,  `st\_ticket\_id` varchar(5) NOT NULL,  `date` date NOT NULL,  PRIMARY KEY (`st\_staff\_id`,`st\_ticket\_id`),  KEY `fk\_sells\_1\_idx` (`st\_ticket\_id`),  CONSTRAINT `fk\_sells\_1` FOREIGN KEY (`st\_ticket\_id`) REFERENCES `ticket` (`ticket\_id`),  CONSTRAINT `fk\_sells\_3` FOREIGN KEY (`st\_staff\_id`) REFERENCES `staff` (`staff\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **22** | **buys** | CREATE TABLE `buys` (  `b\_passenger\_id` varchar(5) NOT NULL,  `b\_ticket\_id` varchar(5) NOT NULL,  `date\_time` varchar(45) DEFAULT NULL,  PRIMARY KEY (`b\_passenger\_id`,`b\_ticket\_id`),  KEY `fk\_buys\_2\_idx` (`b\_ticket\_id`),  CONSTRAINT `fk\_buys\_1` FOREIGN KEY (`b\_passenger\_id`) REFERENCES `a\_class\_passenger` (`passenger\_id`),  CONSTRAINT `fk\_buys\_2` FOREIGN KEY (`b\_ticket\_id`) REFERENCES `ticket` (`ticket\_id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **23** | **Checks\_pass** | CREATE TABLE `checks\_pass` (  `cp\_checker\_id` varchar(5) NOT NULL,  `cp\_pass\_id` varchar(5) NOT NULL,  PRIMARY KEY (`cp\_checker\_id`,`cp\_pass\_id`),  KEY `fk\_checkspass\_1\_idx` (`cp\_pass\_id`),  CONSTRAINT `fk\_checks\_pass\_1` FOREIGN KEY (`cp\_checker\_id`) REFERENCES `ticket\_checker` (`checker\_id`),  CONSTRAINT `fk\_checkspass\_1` FOREIGN KEY (`cp\_pass\_id`) REFERENCES `pass` (`pass\_id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **24** | **Checks\_ticket** | CREATE TABLE `checks\_ticket` (  `ct\_checker\_id` varchar(5) NOT NULL,  `ct\_ticket\_id` varchar(5) NOT NULL,  PRIMARY KEY (`ct\_checker\_id`,`ct\_ticket\_id`),  KEY `fk\_checks\_ticket\_1\_idx` (`ct\_ticket\_id`),  CONSTRAINT `fk\_checks\_ticket\_1` FOREIGN KEY (`ct\_ticket\_id`) REFERENCES `ticket\_checker` (`checker\_id`),  CONSTRAINT `fk\_checks\_ticket\_2` FOREIGN KEY (`ct\_ticket\_id`) REFERENCES `ticket` (`ticket\_id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **25** | **drives** | CREATE TABLE `drives` (  `d\_driver\_id` varchar(5) NOT NULL,  `d\_bus\_no` varchar(7) NOT NULL,  `date` date NOT NULL,  PRIMARY KEY (`d\_driver\_id`,`d\_bus\_no`,`date`),  KEY `fk\_drives\_2\_idx` (`d\_bus\_no`),  CONSTRAINT `fk\_drives\_1` FOREIGN KEY (`d\_driver\_id`) REFERENCES `bus\_driver` (`driver\_id`),  CONSTRAINT `fk\_drives\_2` FOREIGN KEY (`d\_bus\_no`) REFERENCES `bus` (`license\_plate\_no`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **26** | **guest** | CREATE TABLE `guest` (  `guest\_id` varchar(5) NOT NULL,  `g\_a\_star\_id` varchar(5) NOT NULL,  `f\_name` varchar(45) NOT NULL,  `m\_name` varchar(45) DEFAULT NULL,  `l\_name` varchar(45) NOT NULL,  `street` varchar(45) NOT NULL,  `apt\_no` varchar(5) NOT NULL,  `zip\_code` varchar(5) NOT NULL,  `date` date NOT NULL,  `month` varchar(2) NOT NULL,  PRIMARY KEY (`guest\_id`,`g\_a\_star\_id`,`date`),  KEY `fk\_guest\_1\_idx` (`g\_a\_star\_id`),  KEY `fk\_guest\_2\_idx` (`zip\_code`),  CONSTRAINT `fk\_guest\_1` FOREIGN KEY (`g\_a\_star\_id`) REFERENCES `a\_star\_passenger` (`a\_star\_id`),  CONSTRAINT `fk\_guest\_2` FOREIGN KEY (`zip\_code`) REFERENCES `zip\_code` (`zip\_code`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **27** | **travel\_card** | CREATE TABLE `travel\_card` (  `card\_id` varchar(5) NOT NULL,  `card\_a\_star\_id` varchar(5) NOT NULL,  `issue\_date` date NOT NULL,  `expiry\_date` date NOT NULL,  PRIMARY KEY (`card\_id`,`card\_a\_star\_id`),  KEY `fk\_travel\_card\_1\_idx` (`card\_a\_star\_id`),  CONSTRAINT `fk\_travel\_card\_1` FOREIGN KEY (`card\_a\_star\_id`) REFERENCES `a\_star\_passenger` (`a\_star\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **28** | **guest\_phone** | CREATE TABLE `guest\_phone` (  `phn\_guest\_id` varchar(5) NOT NULL,  `phone\_no` varchar(10) NOT NULL,  PRIMARY KEY (`phn\_guest\_id`,`phone\_no`),  CONSTRAINT `fk\_guest\_phone\_1` FOREIGN KEY (`phn\_guest\_id`) REFERENCES `guest` (`guest\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **39** | **promotional\_**  **discount** | CREATE TABLE `promotional\_discount` (  `promo\_id` varchar(5) NOT NULL,  `discount\_percent` int NOT NULL,  `description` varchar(45) NOT NULL,  PRIMARY KEY (`promo\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **30** | **contains** | CREATE TABLE `contains` (  `c\_card\_id` varchar(5) NOT NULL,  `c\_promo\_id` varchar(5) NOT NULL,  `c\_a\_star\_id` varchar(5) NOT NULL,  PRIMARY KEY (`c\_card\_id`,`c\_promo\_id`,`c\_a\_star\_id`),  KEY `fk\_contains\_2\_idx` (`c\_promo\_id`),  KEY `fk\_contains\_3\_idx` (`c\_a\_star\_id`),  CONSTRAINT `fk\_contains\_1` FOREIGN KEY (`c\_card\_id`) REFERENCES `travel\_card` (`card\_id`),  CONSTRAINT `fk\_contains\_2` FOREIGN KEY (`c\_promo\_id`) REFERENCES `promotional\_discount` (`promo\_id`),  CONSTRAINT `fk\_contains\_3` FOREIGN KEY (`c\_a\_star\_id`) REFERENCES `a\_star\_passenger` (`a\_star\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **31** | **can\_have\_**  **a\_class** | CREATE TABLE `can\_have\_a\_class` (  `ch\_pass\_id` varchar(5) NOT NULL,  `ch\_passenger\_id` varchar(5) NOT NULL,  `month` varchar(2) NOT NULL,  PRIMARY KEY (`ch\_pass\_id`,`ch\_passenger\_id`),  KEY `fk\_can\_have\_2\_idx` (`ch\_passenger\_id`),  CONSTRAINT `fk\_can\_have\_1` FOREIGN KEY (`ch\_pass\_id`) REFERENCES `pass` (`pass\_id`),  CONSTRAINT `fk\_can\_have\_2` FOREIGN KEY (`ch\_passenger\_id`) REFERENCES `a\_class\_passenger` (`passenger\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **32** | **Can\_have\_**  **a\_star** | CREATE TABLE `can\_have\_a\_star` (  `ch\_pass\_id` varchar(5) NOT NULL,  `ch\_a\_star\_id` varchar(5) NOT NULL,  `month` varchar(2) NOT NULL,  PRIMARY KEY (`ch\_pass\_id`,`ch\_a\_star\_id`,`month`),  KEY `fk\_can\_have\_a\_star\_2\_idx` (`ch\_a\_star\_id`),  CONSTRAINT `fk\_can\_have\_a\_star\_1` FOREIGN KEY (`ch\_pass\_id`) REFERENCES `pass` (`pass\_id`),  CONSTRAINT `fk\_can\_have\_a\_star\_2` FOREIGN KEY (`ch\_a\_star\_id`) REFERENCES `a\_star\_passenger` (`a\_star\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  |  |  |
| **33** | **is\_a\_class\_**  **a\_star** | CREATE TABLE `is\_a\_class\_a\_star` (  `aa\_passenger\_id` varchar(5) NOT NULL,  `aa\_a\_star\_id` varchar(5) NOT NULL,  PRIMARY KEY (`aa\_passenger\_id`),  KEY `fk\_is\_a\_class\_a\_star\_2\_idx` (`aa\_a\_star\_id`),  CONSTRAINT `fk\_is\_a\_class\_a\_star\_1` FOREIGN KEY (`aa\_passenger\_id`) REFERENCES `a\_class\_passenger` (`passenger\_id`),  CONSTRAINT `fk\_is\_a\_class\_a\_star\_2` FOREIGN KEY (`aa\_a\_star\_id`) REFERENCES `a\_star\_passenger` (`a\_star\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  |  |  |
| **34** | **is\_emp\_a\_star** | CREATE TABLE `is\_emp\_a\_star` (  `ea\_employee\_id` varchar(5) NOT NULL,  `ea\_a\_star\_id` varchar(5) NOT NULL,  PRIMARY KEY (`ea\_employee\_id`),  KEY `fk\_is\_emp\_a\_star\_2\_idx` (`ea\_a\_star\_id`),  CONSTRAINT `fk\_is\_emp\_a\_star\_1` FOREIGN KEY (`ea\_employee\_id`) REFERENCES `employee` (`employee\_id`),  CONSTRAINT `fk\_is\_emp\_a\_star\_2` FOREIGN KEY (`ea\_a\_star\_id`) REFERENCES `a\_star\_passenger` (`a\_star\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |
| **35** | **Zipcode** | CREATE TABLE `zip\_code` (  `zip\_code` varchar(5) NOT NULL,  `city` varchar(45) NOT NULL,  PRIMARY KEY (`zip\_code`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |
|  | | |

**TRIGGERS**

|  |  |
| --- | --- |
| timetable | CREATE DEFINER=`root`@`localhost` TRIGGER `timetable\_BEFORE\_INSERT` BEFORE INSERT ON `timetable` FOR EACH ROW BEGIN  IF NEW.tt\_id REGEXP'^(DT)?[0-9]{2}$' = 0 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Timetable ID must have format`DTXX`';  ELSEIF NEW.day NOT IN ('M', 'T', 'W', 'Th', 'F', 'Sat', 'Sun') THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Day is not correct';  ELSEIF NEW.interval NOT IN ('15', '20', '30') THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Time Interval is not correct, please enter 15, 20 or 30 minute intervals';  END IF;  END |
| CREATE DEFINER=`root`@`localhost` TRIGGER `timetable\_BEFORE\_UPDATE` BEFORE UPDATE ON `timetable` FOR EACH ROW BEGIN  IF NEW.tt\_id REGEXP'^(DT)?[0-9]{2}$' = 0 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Timetable ID must have format`DTXX`';  ELSEIF NEW.day NOT IN ('M', 'T', 'W', 'Th', 'F', 'Sat', 'Sun') THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Day is not correct';  ELSEIF NEW.interval NOT IN ('15', '20', '30') THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Time Interval is not correct, please enter 15, 20 or 30 minute intervals';  END IF;  END |

|  |  |
| --- | --- |
| zipcode | CREATE DEFINER=`root`@`localhost` TRIGGER `zip\_code\_BEFORE\_INSERT` BEFORE INSERT ON `zip\_code` FOR EACH ROW BEGIN  if new.zip\_code regexp '^[0-9]{5}$' = 0 then  signal sqlstate '45000'  set message\_text = 'Zip code format wrong';  END IF;  END |
| CREATE DEFINER=`root`@`localhost` TRIGGER `zip\_code\_BEFORE\_UPDATE` BEFORE UPDATE ON `zip\_code` FOR EACH ROW BEGIN  if new.zip\_code regexp '^[0-9]{5}$' = 0 then  signal sqlstate '45000'  set message\_text = 'Zip code format wrong';  END IF;  END |

|  |  |
| --- | --- |
| person\_phone | CREATE DEFINER=`root`@`localhost` TRIGGER `person\_phone\_BEFORE\_INSERT` BEFORE INSERT ON `person\_phone` FOR EACH ROW BEGIN  if new.phone\_no regexp '^[0-9]{10}$' = 0 then  signal sqlstate '45000'  set message\_text = 'Phone number format is wrong';  END IF;  END |
| CREATE DEFINER=`root`@`localhost` TRIGGER `person\_phone\_BEFORE\_UPDATE` BEFORE UPDATE ON `person\_phone` FOR EACH ROW BEGIN  if new.phone\_no regexp '^[0-9]{10}$' = 0 then  signal sqlstate '45000'  set message\_text = 'Phone number format is wrong';  END IF;  END |

|  |  |
| --- | --- |
| guest\_phone | CREATE DEFINER=`root`@`localhost` TRIGGER `guest\_phone\_BEFORE\_INSERT` BEFORE INSERT ON `guest\_phone` FOR EACH ROW BEGIN  if new.phone\_no regexp '^[0-9]{10}$' = 0 then  signal sqlstate '45000'  set message\_text = 'Phone number format is wrong';  END IF;  END |
| CREATE DEFINER=`root`@`localhost` TRIGGER `guest\_phone\_BEFORE\_UPDATE` BEFORE UPDATE ON `guest\_phone` FOR EACH ROW BEGIN  if new.phone\_no regexp '^[0-9]{10}$' = 0 then  signal sqlstate '45000'  set message\_text = 'Phone number format is wrong';  END IF;  END |

|  |  |
| --- | --- |
| employee | CREATE DEFINER=`root`@`localhost` TRIGGER `employee\_BEFORE\_INSERT` BEFORE INSERT ON `employee` FOR EACH ROW BEGIN  IF new.e\_type not in ('Bus Driver', 'Staff', 'Ticket Checker') then  signal sqlstate '45000'  set message\_text = 'Employee type must be Bus Driver, Staff, Ticket Checker';  END IF;  END |
| CREATE DEFINER=`root`@`localhost` TRIGGER `employee\_BEFORE\_UPDATE` BEFORE UPDATE ON `employee` FOR EACH ROW BEGIN  IF new.e\_type not in ('Bus Driver', 'Staff', 'Ticket Checker') then  signal sqlstate '45000'  set message\_text = 'Employee type must be Bus Driver, Staff, Ticket Checker';  END IF;  END |

|  |  |
| --- | --- |
| promotional\_  discount | CREATE DEFINER=`root`@`localhost` TRIGGER `promotional\_discount\_BEFORE\_INSERT` BEFORE INSERT ON `promotional\_discount` FOR EACH ROW BEGIN  if new.discount\_percent < 0 or new.discount\_percent > 100 then  signal sqlstate '45000'  set message\_text = 'Discount percent out of range';  END IF;  END |
| CREATE DEFINER=`root`@`localhost` TRIGGER `promotional\_discount\_BEFORE\_UPDATE` BEFORE UPDATE ON `promotional\_discount` FOR EACH ROW BEGIN  if new.discount\_percent < 0 or new.discount\_percent > 100 then  signal sqlstate '45000'  set message\_text = 'Discount percent out of range';  END IF;  END |

|  |  |
| --- | --- |
| payment\_details | CREATE DEFINER=`root`@`localhost` TRIGGER `payment\_details\_BEFORE\_INSERT` BEFORE INSERT ON `payment\_details` FOR EACH ROW BEGIN  IF NEW.method not in ('cash', 'card') THEN  SIGNAL sqlstate '45000'  SET message\_text = 'Payment method is wrong';  END IF;  END |
| CREATE DEFINER=`root`@`localhost` TRIGGER `payment\_details\_BEFORE\_UPDATE` BEFORE UPDATE ON `payment\_details` FOR EACH ROW BEGIN  IF NEW.method not in ('cash', 'card') THEN  SIGNAL sqlstate '45000'  SET message\_text = 'Payment method is wrong';  END IF;  END |

|  |  |
| --- | --- |
| person | CREATE DEFINER=`root`@`localhost` TRIGGER `person\_check\_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 of person must be greater than 16 years.';  ELSEIF NEW.person\_id REGEXP'^[P][0-9]{3}$' = 0 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'PersonID must have format`PXXX`';  ELSEIF NEW.gender NOT IN ('M', 'F') THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Gender is not correct.';  END IF;  END |
| CREATE DEFINER=`root`@`localhost` TRIGGER `person\_check\_before\_update` BEFORE UPDATE ON `person` FOR EACH ROW BEGIN  IF TIMESTAMPDIFF(YEAR, NEW.dob, CURDATE()) < 16 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Age of person must be greater than 16 years.';  ELSEIF NEW.person\_id REGEXP'^[P][0-9]{3}$' = 0 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'PersonID must have format`PXXX`';  ELSEIF NEW.gender NOT IN ('M', 'F') THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Gender is not correct.';  END IF;  END |

|  |  |
| --- | --- |
| bus | CREATE DEFINER=`root`@`localhost` TRIGGER `bus\_BEFORE\_INSERT` BEFORE INSERT ON `bus` FOR EACH ROW BEGIN  IF new.license\_plate\_no regexp '^[A-Z]{3}[0-9]{4}$' = 0 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'License plate number format is incorrect';  ELSEIF new.no\_of\_seats > 100 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Number of seats cannot be more than 100';  END IF;  END |
| CREATE DEFINER=`root`@`localhost` TRIGGER `bus\_BEFORE\_UPDATE` BEFORE UPDATE ON `bus` FOR EACH ROW BEGIN  IF new.license\_plate\_no regexp '^[A-Z]{3}[0-9]{4}$' = 0 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'License plate number format is incorrect';  ELSEIF new.no\_of\_seats > 100 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Number of seats cannot be more than 100';  END IF;  END |

|  |  |
| --- | --- |
| guest | CREATE DEFINER=`root`@`localhost` TRIGGER `guest\_BEFORE\_INSERT` BEFORE INSERT ON `guest` FOR EACH ROW BEGIN  IF (select COUNT(\*) from guest where g\_a\_star\_id = new.g\_a\_star\_id and `month` = new.`month`) = 4 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'A-Star Passenger cannot have more than 4 guests in a month';  ELSEIF new.`month` regexp '^[0][1-9]$' = 0 THEN  IF new.`month` regexp '^[1][0-2]$' = 0 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Month format is incorrect';  END IF;  END IF;  END |
| CREATE DEFINER=`root`@`localhost` TRIGGER `guest\_BEFORE\_UPDATE` BEFORE UPDATE ON `guest` FOR EACH ROW BEGIN  IF(select COUNT(\*) from guest where g\_a\_star\_id = new.g\_a\_star\_id and `month` = new.`month`) = 4 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'A-Star Passenger cannot have more than 4 guests in a month';  ELSEIF new.`month` regexp '^[0][1-9]$' = 0 THEN  IF NEW.`month` regexp '^[1][0-2]$' = 0 THEN  SIGNAL SQLSTATE '45000'  SET message\_text = 'Month format is incorrect';  END IF;  END IF;  END |

**Phase III. d. Use the Create View statement to create the following 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\_a\_star\_passengers` AS  select person\_id, f\_name as first\_name, l\_name as last\_name, issue\_date as date\_of\_membership  from person, ticket, travel\_card  where (person\_id, card\_id) in  (select person\_id, card\_id  from person, travel\_card  where exists  (select \*  from a\_class\_passenger, is\_a\_class\_a\_star  where passenger\_id = aa\_passenger\_id and person\_id = ac\_person\_id and aa\_a\_star\_id = card\_a\_star\_id)  or exists  (select \*  from employee, is\_emp\_a\_star  where employee\_id = ea\_employee\_id and person\_id = emp\_person\_id and ea\_a\_star\_id = card\_a\_star\_id))  and person\_id = t\_person\_id  and date > date(current\_date - interval 1 month)  group by t\_person\_id  having count(ticket\_id) > 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 license\_plate\_no, bus\_no, no\_of\_seats, bus\_type  from bus, ticket  where license\_plate\_no = t\_bus\_no  and date > date(current\_date - interval 2 month)  group by t\_bus\_no  order by count(ticket\_id)  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\_a\_star\_passenger` AS  select f\_name as first\_name, m\_name as middle\_name, l\_name as last\_name, phone\_no as phone\_number, person\_id  from person, person\_phone, ticket  where not exists  (select \*  from a\_class\_passenger, is\_a\_class\_a\_star  where passenger\_id = aa\_passenger\_id and person\_id = ac\_person\_id)  and person\_id = phn\_person\_id  and person\_id = t\_person\_id  and date > date(current\_date - interval 2 month)  group by t\_person\_id  having count(ticket\_id) > 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 f\_name as first\_name, l\_name as last\_name, start\_date, e\_type as employee\_type  from person, employee, ticket  where person\_id = emp\_person\_id  and person\_id = t\_person\_id  and date > date(current\_date - interval 1 month)  group by t\_person\_id  having max(ticket\_id); |

**Phase III. e. Answer the following Queries. Feel free to use any of the views that you created in part (d.):**

|  |  |
| --- | --- |
| **1** | **For each employee class, list the employees belonging to that class.** |
|  | select e.e\_type, e.employee\_id, p.f\_name, p.l\_name, p.gender from employee e, person p where p.person\_id = e.emp\_person\_id ORDER BY e.e\_type; |
|  | |
| **2** | **Find the names of employees who are also an A-Class Passenger.** |
|  | select p.person\_id, p.f\_name, p.m\_name, p.l\_name FROM employee e, person p, a\_class\_passenger a where e.emp\_person\_id = a.ac\_person\_id AND e.emp\_person\_id = p.person\_id; |
|  | |
| **3** | **Find the average number of bookings made by the top five A-Star Passengers.** |
|  | select avg(count) as avg\_no\_bookings from (select count(ticket\_id) as count from top\_a\_star\_passengers, ticket where person\_id = t\_person\_id group by t\_person\_id order by count(ticket\_id) limit 5) as bookings; |
|  | |
| **4** | **Find the Bus ID and Route names of the bus that is booked the most.** |
|  | select b.license\_plate\_no as bus\_id, b.bus\_route\_id as route\_id from popular\_bus as p, bus as b where p.license\_plate\_no = b.license\_plate\_no; |
|  | |
| **5** | **Find Bus ID that has been cancelled more than 3 times in the past month.** |
|  | NOT APPLICABLE |
|  | |
| **6** | **Find the total number bookings for each bus in the system.** |
|  | SELECT t.t\_bus\_no, COUNT(t.t\_bus\_no) FROM ticket t GROUP BY t.t\_bus\_no; |
|  | |
| **7** | **Find the driver details who has driven every day of the past week.** |
|  | select distinct f\_name as first\_name, l\_name as last\_name, dob as date\_of\_birth, street, city, p.zip\_code as zip\_code from person as p, zip\_code as z, employee, bus\_driver, drives where d\_driver\_id = driver\_id and d\_employee\_id = employee\_id and emp\_person\_id = person\_id and p.zip\_code = z.zip\_code and date > date(current\_date - interval 7 day) having count(d\_driver\_id) = 7; |
|  | |
| **8** | **Find the count of passengers who booked the most popular bus.** |
|  | select count(t\_person\_id) from ticket where t\_bus\_no = (select license\_plate\_no from popular\_bus) group by t\_bus\_no; |
|  | |
| **9** | **List all the booking details issued after the most current employee was hired.** |
|  | select \* from ticket where date > (select max(start\_date) from employee); |
|  | |
| **10** | **List all the employees that have enrolled as A-Star Passengers within a month of being employed.** |
|  | select f\_name, l\_name, e\_type from person, employee, is\_emp\_a\_star, travel\_card where person\_id = emp\_person\_id and employee\_id = ea\_employee\_id and ea\_a\_star\_id = card\_a\_star\_id and issue\_date < date(start\_date + interval 1 month); |
|  | |
| **11** | **Find the route with the highest number of bus stops.** |
|  | SELECT stop\_route\_id, COUNT(\*) total FROM bus\_stop GROUP BY stop\_route\_id ORDER BY COUNT(\*) DESC LIMIT 1; |
|  | |
| **12** | **Find the name of passengers who have been A-Star Passengers for over 5 years.** |
|  | select f\_name, l\_name from person, a\_class\_passenger, is\_a\_class\_a\_star, travel\_card where person\_id = ac\_person\_id and passenger\_id = aa\_passenger\_id and aa\_a\_star\_id = card\_a\_star\_id and issue\_date > date(current\_date - interval 5 year); |
|  | |
| **13** | **Find the bookings made by the potential A-Star Passengers in the last year.** |
|  | select \* from ticket where t\_person\_id in (select person\_id from potential\_a\_star\_passenger) and date > date(current\_date - interval 1 year); |
|  | |