بسم الله الرحمن الرحيم

Project name: Metro

Project Code : MIAM

Members name :

 1. Ahmed Yasiin Ashraf Ibrahim Eldsoky.
 2. Mohamed Ahmed Qorany Ewis.
 3. Mohamed Farag Mohamed Hafez.

 Supervisor name:

 ➢ Ibrahim Ashraf Abdelraouf Hassan.

Problem statement and the importance of our database:

i.   Problem statement :

1. **Efficient Metro Operations:** The database could help manage and optimize the operation of the metro system, including information about stations, trains, schedules, and employees.
2. **Passenger Management:** The system may help keep track of passengers, their personal information, travel history, and the types of access methods they use (tickets, cards, subscriptions).
3. **Safety and Accident Management:** With details about safety services, accidents, and occurrences, the database can contribute to the overall safety and security of the metro system. It allows for efficient tracking and resolution of incidents.

4. **Employee Management:** The system provides information about metro employees, including their roles, stations they manage, and their work details.
5. **Financial Transactions:** The database may facilitate financial transactions related to ticket purchases, subscriptions, and other services, contributing to revenue tracking and financial management.

ii. Importance of our database :

1. **Data Centralization:** The database serves as a central repository for various aspects of the metro system, allowing for organized and centralized data management.
2. **Efficiency and Productivity:** Efficiently managing metro operations and having a comprehensive view of passenger data can contribute to increased efficiency and productivity.
3. **Safety and Incident Response:** The database's ability to store safety-related information facilitates a quick response to incidents, ensuring the safety of passengers and staff.
4. **Strategic Decision-Making:** Access to historical data and real-time information allows metro authorities to make informed decisions, optimize schedules, and improve overall performance.
5. **Customer Experience:** Detailed information about passengers and their preferences enables the metro system to tailor services, improving the overall customer experience.
6. **Regulatory Compliance:** The database can assist in ensuring compliance with regulations, especially in safety and security matters.
7. **Financial Management:** Tracking financial transactions and revenue can contribute to better financial management and planning.
8. **Maintenance and Planning:** Information about trains, stations, and incidents can aid in maintenance planning, ensuring the smooth functioning of the metro system.

Requirements :

1. **Passengers and Tickets:**
   - How are passenger data recorded? Is there a need to store personal information?
   - How are tickets issued? Is there a payment system dealing with cash or credit cards?
   - Are there subscription systems for frequent passengers?

2. **Stations and Schedules:**
   - How is station data stored? Are there geographical location details?
   - Are there timetables for train trips? How can this data be efficiently recorded?

3. **Employees:**
   - How is employee information recorded? Are there different categories of employees?
   - How is data about tasks and responsibilities stored?
   - Are there schedules for employee patrols or shifts?

4. **Safety and Accidents:**
   - How can safety and incident data be accurately recorded?
   - Are there procedures for tracking emergencies and responding to them?
   - Are there periodic reports or analyses of past incidents?

5. **Line and transformation:**
   - The system should allow efficient and flexible definition and design of lines, with the ability to set various properties for each line, such as color and type.
   - The system should efficiently manage lines, enabling users to add, delete, and modify lines based on their needs.
   - The system should provide ease for making transformations on the lines, allowing for easy changes to their direction.

6. Access Method As Tickets and Card:

- There needs to be a mechanism for validating tickets at entry points, ensuring that only valid and non-expired tickets are accepted.
- The database should manage the issuance of metro cards, storing relevant user information and associating it with each card.

EER digram :
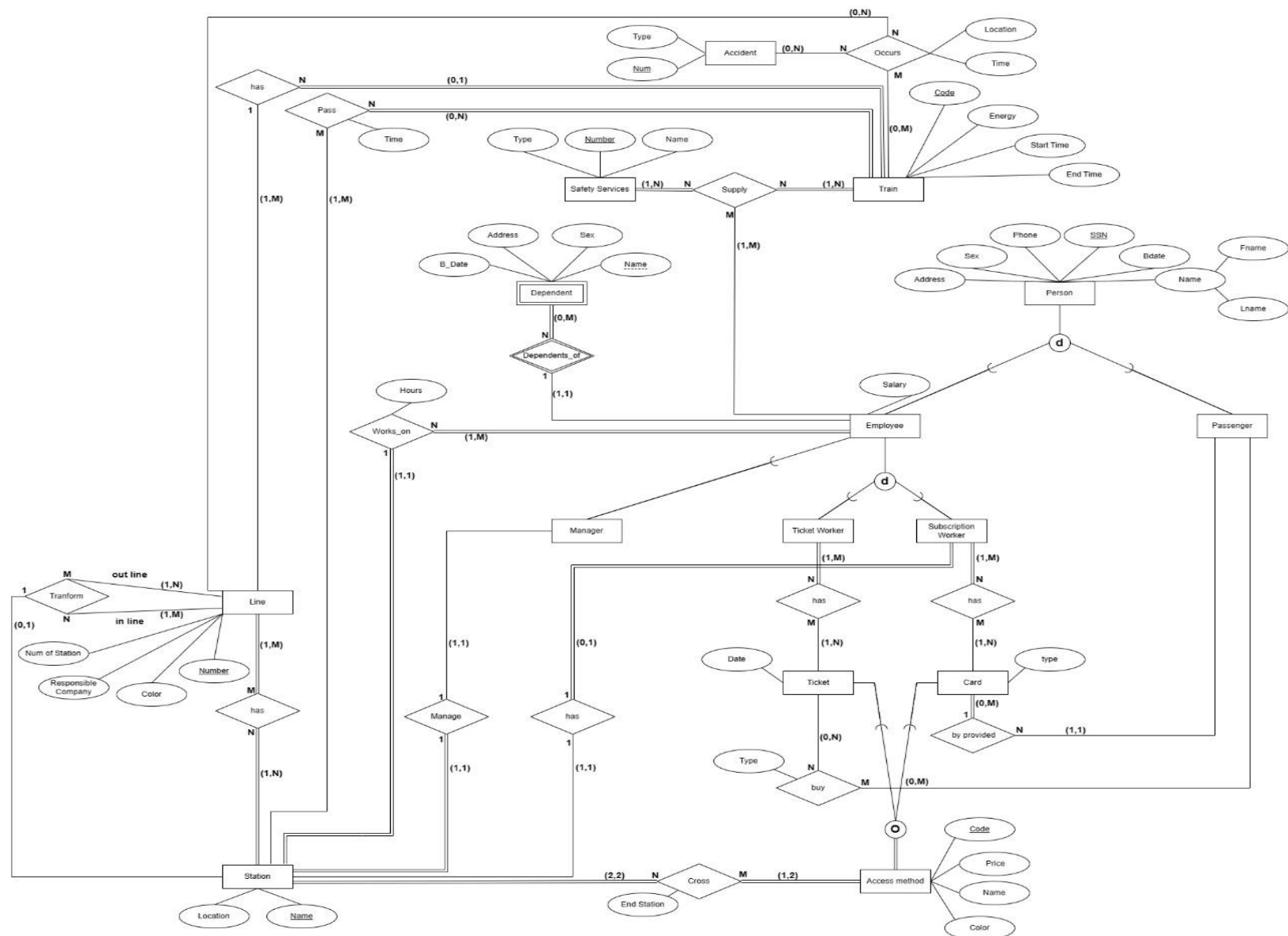
Relation model :

Person : (SSN , Fname , Lname , Address , Sex , Bdate , phone)

Manger : (Pssn)

Employee : (Pssn , salary , Sname , Hours)

                NN (above Sname)

Passenger : (Pssn , Acode)

Dependent : (Pssn , Name , Sex , Address , Bdate)

Ticket worker : (Pssn)

Subscription worker : (Pssn , Sname)

Access method : (Code , Type_method , Price , Tflag , Date , Cflag , Type)

T_has : (Pssn , Tcode)
             NN (above Pssn)

C_has : (Pssn , Ccode)
             NN (above Pssn)

Buy : (Tcode , Pssn , Type)

Crosss : (Acode , Sname , End-station)
        NN   NN (above Acode, Sname)

Station : (<u>Name</u> , location , pssn ^NN)

Line : (<u>Number</u> , Color , Responsible-company , <u>Num-of-station</u>)

Transform : (<u>Sname</u> , <u>In-Lnumber</u> , <u>Out-Lnumber</u>)

L_has : (<u>Sname</u> ^NN , <u>Lnumber</u> ^NN)

Train : (<u>Code</u> , Energy , Start-time , End-time , Lnumber)

Safety services : (<u>SsNumber</u> , Name , Type)

Tr_Pass : (<u>Tr-code</u> , <u>Sname</u> ^NN , Time)

Supply : (<u>Tr-code</u> , safety-number ^NN , Pssn ^NN)

Accident : (<u>Num</u> , Type)

Occurs : (<u>Tr-code</u> , <u>Anum</u> , <u>Lnumber</u> , Time , Location)

Sql of creation :

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

```sql
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema Metro

-- -----------------------------------------------------

DROP SCHEMA IF EXISTS `Metro` ;

-- -----------------------------------------------------

-- Schema Metro

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `Metro` DEFAULT CHARACTER SET utf8 ;

USE `Metro` ;

-- -----------------------------------------------------

-- Table `Metro`.`Person`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Metro`.`Person` ;

CREATE TABLE IF NOT EXISTS `Metro`.`Person` (

  `ssn` VARCHAR(50) NOT NULL,

  `Fname` VARCHAR(20) NULL,

  `Lname` VARCHAR(20) NULL,

  `Address` VARCHAR(40) NULL,

  `Sex` ENUM('M', 'F') NULL,

  `Bdate` DATE NULL,
```

```sql
  `phone` VARCHAR(45) NULL,
  PRIMARY KEY (`ssn`))
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Metro`.`Manger`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`Manger` ;
CREATE TABLE IF NOT EXISTS `Metro`.`Manger` (
  `Pssn` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`Pssn`),
  CONSTRAINT `fk_Manger_Person1`
    FOREIGN KEY (`Pssn`)
    REFERENCES `Metro`.`Person` (`ssn`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Metro`.`Station`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`Station` ;
CREATE TABLE IF NOT EXISTS `Metro`.`Station` (
  `Name` VARCHAR(30) NOT NULL,
  `Location` VARCHAR(40) NOT NULL,
```

```sql
  `Mpssn` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`Name`),
  INDEX `fk_Station_Manger1_idx` (`Mpssn` ASC),
  CONSTRAINT `fk_Station_Manger1`
    FOREIGN KEY (`Mpssn`)
    REFERENCES `Metro`.`Manger` (`Pssn`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Metro`.`Employee`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`Employee` ;
CREATE TABLE IF NOT EXISTS `Metro`.`Employee` (
  `Pssn` VARCHAR(50) NOT NULL,
  `Sname` VARCHAR(30) NOT NULL,
  `salary` DOUBLE NOT NULL,
  `hours` DOUBLE NOT NULL,
  PRIMARY KEY (`Pssn`),
  INDEX `fk_Employee_Station1_idx` (`Sname` ASC),
  CONSTRAINT `fk_Employee_Person1`
    FOREIGN KEY (`Pssn`)
    REFERENCES `Metro`.`Person` (`ssn`)
```

```
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Employee_Station1`
    FOREIGN KEY (`Sname`)
    REFERENCES `Metro`.`Station` (`Name`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-- -------------------------------------------------
-- Table `Metro`.`Access_Method`
-- -------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`Access_Method` ;
CREATE TABLE IF NOT EXISTS `Metro`.`Access_Method` (
  `Code` VARCHAR(20) NOT NULL,
  `name` VARCHAR(20) NOT NULL,
  `Price` INT NOT NULL,
  `Tflag` VARCHAR(10) NOT NULL,
  `Date` DATE NULL,
  `Cflag` VARCHAR(10) NOT NULL,
  `Type` VARCHAR(40) NULL,
  PRIMARY KEY (`Code`))
ENGINE = InnoDB;
-- -------------------------------------------------
```

```sql
-- Table `Metro`.`Passenger`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`Passenger` ;

CREATE TABLE IF NOT EXISTS `Metro`.`Passenger` (
  `Pssn` VARCHAR(50) NOT NULL,
  `code` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`Pssn`),
  INDEX `fk_Passenger_Access-Method1_idx` (`code` ASC),
  CONSTRAINT `fk_Passenger_Person1`
    FOREIGN KEY (`Pssn`)
    REFERENCES `Metro`.`Person` (`ssn`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Passenger_Access-Method1`
    FOREIGN KEY (`code`)
    REFERENCES `Metro`.`Access_Method` (`Code`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Metro`.`Dependent`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`Dependent` ;
```

```sql
CREATE TABLE IF NOT EXISTS `Metro`.`Dependent` (

  `name` VARCHAR(20) NOT NULL,

  `Sex` ENUM('M', 'F') NOT NULL,

  `Address` VARCHAR(45) NOT NULL,

  `Bdate` DATE NOT NULL,

  `Pssn` VARCHAR(50) NULL,

  PRIMARY KEY (`name`, `Pssn`),

  INDEX `fk_Dependent_Employee1_idx` (`Pssn` ASC),

  CONSTRAINT `fk_Dependent_Employee1`

    FOREIGN KEY (`Pssn`)

    REFERENCES `Metro`.`Employee` (`Pssn`)

    ON DELETE CASCADE

    ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------

-- Table `Metro`.`Ticket_worker`

-- -----------------------------------------------

DROP TABLE IF EXISTS `Metro`.`Ticket_worker` ;

CREATE TABLE IF NOT EXISTS `Metro`.`Ticket_worker` (

  `Pssn` VARCHAR(50) NOT NULL,

  PRIMARY KEY (`Pssn`),

  CONSTRAINT `fk_Ticket-worker_Person`

    FOREIGN KEY (`Pssn`)
```

```
    REFERENCES `Metro`.`Person` (`ssn`)

    ON DELETE CASCADE

    ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------

-- Table `Metro`.`Subscription_worker`

-- -----------------------------------------------

DROP TABLE IF EXISTS `Metro`.`Subscription_worker` ;

CREATE TABLE IF NOT EXISTS `Metro`.`Subscription_worker` (

  `Pssn` VARCHAR(50) NOT NULL,

  `SName` VARCHAR(30) NULL,

  PRIMARY KEY (`Pssn`),

  INDEX `fk_Subscription-worker_Station1_idx` (`SName` ASC),

  CONSTRAINT `fk_Subscription-worker_Person1`

    FOREIGN KEY (`Pssn`)

    REFERENCES `Metro`.`Person` (`ssn`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  CONSTRAINT `fk_Subscription-worker_Station1`

    FOREIGN KEY (`SName`)

    REFERENCES `Metro`.`Station` (`Name`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)
```

```sql
ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Metro`.`Line`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Metro`.`Line` ;

CREATE TABLE IF NOT EXISTS `Metro`.`Line` (

  `Number` INT NOT NULL,

  `Color` VARCHAR(10) NOT NULL,

  `Responsible-company` VARCHAR(40) NOT NULL,

  `Num-of-Station` INT NOT NULL,

  PRIMARY KEY (`Number`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Metro`.`Train`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Metro`.`Train` ;


CREATE TABLE IF NOT EXISTS `Metro`.`Train` (

  `Code` VARCHAR(20) NOT NULL,

  `Energy` ENUM('FULL', 'MEDIUM', 'LOW') NOT NULL,

  `Start-time` TIME NOT NULL,

  `End-time` TIME NOT NULL,

  `LNumber` INT NULL,
```

```sql
  PRIMARY KEY (`Code`),

  INDEX `fk_Train_Line1_idx` (`LNumber` ASC),

  CONSTRAINT `fk_Train_Line1`

    FOREIGN KEY (`LNumber`)

    REFERENCES `Metro`.`Line` (`Number`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------

-- Table `Metro`.`Safety_services`

-- -----------------------------------------------

DROP TABLE IF EXISTS `Metro`.`Safety_services` ;

CREATE TABLE IF NOT EXISTS `Metro`.`Safety_services` (

  `SsNumber` INT NOT NULL,

  `Name` VARCHAR(20) NOT NULL,

  `Type` VARCHAR(45) NOT NULL,

  PRIMARY KEY (`SsNumber`))

ENGINE = InnoDB;

-- -----------------------------------------------

-- Table `Metro`.`Accident`

-- -----------------------------------------------

DROP TABLE IF EXISTS `Metro`.`Accident` ;
```

```sql
CREATE TABLE IF NOT EXISTS `Metro`.`Accident` (

  `Num` INT NOT NULL,

  `Type` VARCHAR(40) NOT NULL,

  PRIMARY KEY (`Num`))

ENGINE = InnoDB;

-- --------------------------------------------------

-- Table `Metro`.`T_has`

-- --------------------------------------------------

DROP TABLE IF EXISTS `Metro`.`T_has` ;

CREATE TABLE IF NOT EXISTS `Metro`.`T_has` (

  `TCode` VARCHAR(20) NOT NULL,

  `Pssn` VARCHAR(50) NOT NULL,

  PRIMARY KEY (`TCode`, `Pssn`),

  INDEX `fk_Access-Method_has_Ticket-worker_Ticket-worker1_idx` (`Pssn` ASC),

  INDEX `fk_Access-Method_has_Ticket-worker_Access-Method1_idx` (`TCode` ASC),

  CONSTRAINT `fk_Access-Method_has_Ticket-worker_Access-Method1`

    FOREIGN KEY (`TCode`)

    REFERENCES `Metro`.`Access_Method` (`Code`)

    ON DELETE CASCADE

    ON UPDATE CASCADE,

  CONSTRAINT `fk_Access-Method_has_Ticket-worker_Ticket-worker1`
```

```
    FOREIGN KEY (`Pssn`)
    REFERENCES `Metro`.`Ticket_worker` (`Pssn`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Metro`.`C_has`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`C_has` ;

CREATE TABLE IF NOT EXISTS `Metro`.`C_has` (
  `Ccode` VARCHAR(20) NOT NULL,
  `pssn` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`Ccode`, `pssn`),
  INDEX `fk_Access-Method_has_Subscription-worker_Subscription-worke_idx` (`pssn` ASC),
  INDEX `fk_Access-Method_has_Subscription-worker_Access-Method1_idx` (`Ccode` ASC),
  CONSTRAINT `fk_Access-Method_has_Subscription-worker_Access-Method1`
    FOREIGN KEY (`Ccode`)
    REFERENCES `Metro`.`Access_Method` (`Code`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
```

```
  CONSTRAINT `fk_Access-Method_has_Subscription-worker_Subscription-
worker1`

    FOREIGN KEY (`pssn`)

    REFERENCES `Metro`.`Subscription_worker` (`Pssn`)

    ON DELETE CASCADE

    ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Metro`.`Buy`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `Metro`.`Buy` ;

CREATE TABLE IF NOT EXISTS `Metro`.`Buy` (

  `Tcode` VARCHAR(20) NULL,

  `Pssn` VARCHAR(50) NULL,

  `Type` VARCHAR(10) NOT NULL,

  PRIMARY KEY (`Tcode`, `Pssn`),

  INDEX `fk_Passenger_has_Access-Method_Access-Method1_idx` (`Tcode`
ASC),

    INDEX `fk_Passenger_has_Access-Method_Passenger1_idx` (`Pssn` ASC),

  CONSTRAINT `fk_Passenger_has_Access-Method_Passenger1`

    FOREIGN KEY (`Pssn`)

    REFERENCES `Metro`.`Passenger` (`Pssn`)

    ON DELETE CASCADE

    ON UPDATE CASCADE,
```

```sql
  CONSTRAINT `fk_Passenger_has_Access-Method_Access-Method1`
    FOREIGN KEY (`Tcode`)
    REFERENCES `Metro`.`Access_Method` (`Code`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Metro`.`Crosss`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`Crosss` ;

CREATE TABLE IF NOT EXISTS `Metro`.`Crosss` (
  `ACode` VARCHAR(20) NOT NULL,
  `SName` VARCHAR(30) NOT NULL,
  `End_station` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`ACode`, `SName`),
  INDEX `fk_Access-Method_has_Station_Station1_idx` (`SName` ASC),
  INDEX `fk_Access-Method_has_Station_Access-Method1_idx` (`ACode` ASC),
  CONSTRAINT `fk_Access-Method_has_Station_Access-Method1`
    FOREIGN KEY (`ACode`)
    REFERENCES `Metro`.`Access_Method` (`Code`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
```

```sql
  CONSTRAINT `fk_Access-Method_has_Station_Station1`
    FOREIGN KEY (`SName`)
    REFERENCES `Metro`.`Station` (`Name`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Metro`.`L_has`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`L_has` ;

CREATE TABLE IF NOT EXISTS `Metro`.`L_has` (
  `SName` VARCHAR(30) NOT NULL,
  `LNumber` INT NOT NULL,
  PRIMARY KEY (`SName`, `LNumber`),
  INDEX `fk_Station_has_Line_Line1_idx` (`LNumber` ASC),
  INDEX `fk_Station_has_Line_Station1_idx` (`SName` ASC),
  CONSTRAINT `fk_Station_has_Line_Station1`
    FOREIGN KEY (`SName`)
    REFERENCES `Metro`.`Station` (`Name`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Station_has_Line_Line1`
    FOREIGN KEY (`LNumber`)
```

```
    REFERENCES `Metro`.`Line` (`Number`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-- -------------------------------------------------
-- Table `Metro`.`transform`
-- -------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`transform` ;
CREATE TABLE IF NOT EXISTS `Metro`.`transform` (
  `in_LNumber` INT NOT NULL,
  `out_LNumber` INT NOT NULL,
  `SName` VARCHAR(30) NULL,
  PRIMARY KEY (`in_LNumber`, `out_LNumber`, `SName`),
  INDEX `fk_Line_has_Line_Line2_idx` (`in_LNumber` ASC),
  INDEX `fk_Line_has_Line_Line1_idx` (`out_LNumber` ASC),
  INDEX `fk_Line_has_Line_Station1_idx` (`SName` ASC),
  CONSTRAINT `fk_Line_has_Line_Line1`
    FOREIGN KEY (`out_LNumber`)
    REFERENCES `Metro`.`Line` (`Number`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Line_has_Line_Line2`
    FOREIGN KEY (`in_LNumber`)
```

```
    REFERENCES `Metro`.`Line` (`Number`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Line_has_Line_Station1`
    FOREIGN KEY (`SName`)
    REFERENCES `Metro`.`Station` (`Name`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Metro`.`Tr_pass`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`Tr_pass` ;
CREATE TABLE IF NOT EXISTS `Metro`.`Tr_pass` (
  `TrCode` VARCHAR(20) NOT NULL,
  `SName` VARCHAR(30) NULL,
  `time` TIME NOT NULL,
  PRIMARY KEY (`TrCode`, `SName`),
  INDEX `fk_Train_has_Station_Station1_idx` (`SName` ASC),
  INDEX `fk_Train_has_Station_Train1_idx` (`TrCode` ASC),
  CONSTRAINT `fk_Train_has_Station_Train1`
    FOREIGN KEY (`TrCode`)
    REFERENCES `Metro`.`Train` (`Code`)
```

```
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Train_has_Station_Station1`
    FOREIGN KEY (`SName`)
    REFERENCES `Metro`.`Station` (`Name`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Metro`.`supply`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`supply` ;

CREATE TABLE IF NOT EXISTS `Metro`.`supply` (
  `Safety_Number` INT NOT NULL,
  `TrCode` VARCHAR(20) NOT NULL,
  `Pssn` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`Safety_Number`, `TrCode`, `Pssn`),
  INDEX `fk_Train_has_Safety-services_Safety-services1_idx`
(`Safety_Number` ASC),
  INDEX `fk_Train_has_Safety-services_Train1_idx` (`TrCode` ASC),
  INDEX `fk_Train_has_Safety-services_Employee1_idx` (`Pssn` ASC),
  CONSTRAINT `fk_Train_has_Safety-services_Train1`
    FOREIGN KEY (`TrCode`)
```

```
    REFERENCES `Metro`.`Train` (`Code`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Train_has_Safety-services_Safety-services1`
    FOREIGN KEY (`Safety_Number`)
    REFERENCES `Metro`.`Safety_services` (`SsNumber`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Train_has_Safety-services_Employee1`
    FOREIGN KEY (`Pssn`)
    REFERENCES `Metro`.`Employee` (`Pssn`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `Metro`.`occurs`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Metro`.`occurs` ;

CREATE TABLE IF NOT EXISTS `Metro`.`occurs` (
  `ANum` INT NULL,
  `TrCode` VARCHAR(20) NULL,
  `LNumber` INT NULL,
  `time` TIME NOT NULL,
```

```
  `location` VARCHAR(44) NOT NULL,
  PRIMARY KEY (`ANum`, `TrCode`, `LNumber`),
  INDEX `fk_Accident_has_Train_Train1_idx` (`TrCode` ASC),
  INDEX `fk_Accident_has_Train_Accident1_idx` (`ANum` ASC),
  INDEX `fk_Accident_has_Train_Line1_idx` (`LNumber` ASC),
  CONSTRAINT `fk_Accident_has_Train_Accident1`
    FOREIGN KEY (`ANum`)
    REFERENCES `Metro`.`Accident` (`Num`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Accident_has_Train_Train1`
    FOREIGN KEY (`TrCode`)
    REFERENCES `Metro`.`Train` (`Code`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Accident_has_Train_Line1`
    FOREIGN KEY (`LNumber`)
    REFERENCES `Metro`.`Line` (`Number`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
```

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

Inserted data:

INSERT INTO line

VALUES(1, 'blue', 44, 35),

(2, 'red', 21.6, 20),

(3, 'green', 41.2, 34);

-- ------------

INSERT INTO train

VALUES('10642LG', 'low', '16:15:39', '22:11:37',1),

('25635KQ', 'MEDIUM', '10:09:41', '08:39:41',3),

('22394RR', 'MEDIUM', '10:25:15', '18:26:27',2),

('16278MG', 'full', '10:24:42', '06:02:46',2),

('27531JI', 'full', '20:45:44', '14:11:28',1),

('19053KT', 'low', '17:36:24', '05:10:53',3);

-- -------------

INSERT INTO person

VALUES(9965412365, 'mohamed' , 'eldsoky' , 'el-mahdi' , 'M' , '1942-04-23' , '01512814'),

(2111137605, 'ibrahim' , 'eldsoky' , 'nour-el-deen' , 'M' , '1970-02-27' , '01128218'),

(3111143747, 'shahd' , 'ali' , 'al-bashir' , 'F' , '1948-07-22' , '0117634'),

(4111113930, 'Gharib' , 'mosa' , 'el-khalifa-el-maamoun' , 'M' , '1945-09-14' , '01129950'),

(5111123090, 'ashraf' , 'nader' , 'nour-el-deen' , 'M' , '2022-10-9' , '011387'),

(6111137166, 'ashraf' , 'rashad' , 'el-khalifa-el-maamoun' , 'M' , '1990-01-3' , '0117775'),

(7111125839, 'taher' , 'kmal' , 'nour-el-deen' , 'M' , '1993-02-11' , '01127710'),

(8111141088, 'ashraf' , 'mansor' , 'el-mahdi' , 'M' , '1953-02-02' , '01117895'),

(9111142622, 'ayaat' , 'ramy' , 'mahmoud-el-attar' , 'F' , '2009-01-22' , '0112564'),

(1011137605, 'Ghanem' , 'saber' , 'mahmoud-el-attar' , 'M' , '1994-10-14' , '01128669'),

(1111139227, 'mazen' , 'ibrahim' , 'al-bashir' , 'M' , '1966-05-04' , '011954'),

(1211136745, 'ashraf' , 'mohamed' , 'el-mahdi' , 'M' , '2002-03-24' , '01125734'),

(1311136061, 'ibrahim' , 'Ghanem' , 'mahmoud-el-attar' , 'M' , '1966-05-27' , '0113439'),

(1411116284, 'ahmad' , 'ibrahim' , 'nour-el-deen' , 'M' , '1977-12-23' , '01120942'),

(1511134012, 'rahma' , 'mohamed' , 'el-mahdi' , 'F' , '2007-07-11' , '01126339'),

(1611117300, 'mazen' , 'kareem' , 'el-khalifa-el-maamoun' , 'M' , '1983-09-18' , '01023464'),

(1711142801, 'rahma' , 'mazen' , 'mahmoud-el-attar' , 'F' , '1951-12-02' , '01121062'),

(1811131612, 'mohamed' , 'mazen' , 'el-khalifa-el-maamoun' , 'M' , '2007-12-16' , '01215058'),

(1911142470, 'Ghanem' , 'eldsoky' , 'nour-el-deen' , 'M' , '1979-11-15' , '01217726'),

```sql
(2011133463, 'yasiin' , 'said' , 'el-mahdi' , 'M' , '1983-12-16' , '01120636'),

(2145678945, 'ibrahim' , 'ashraf' , 'abdelraouf' , 'M' , '2004-12-16' ,
'01020636'),

(2245678978, 'mohamed' , 'ahmed' , 'el-mahdi' , 'M' , '2006-09-08' ,
'01093645'),

(2345698796, 'ahmad' , 'eldsoky' , 'mogawra40' , 'M' , '1907-03-30' ,
'0124485054'),

(2432178965, 'adel' , 'bola' , 'mogawra45' , 'M' , '1988-08-19' , '012554984'),

(2512365478, 'sama' , 'eldsoky' , 'mogawra40' , 'F' , '1906-09-10' ,
'01049464'),

(2665478925, 'lolo' , 'tarek' , 'alahram' , 'F' , '1907-07-9' , '01545125'),

(2765478925, 'ali' , 'mohamed' , 'alahram' , 'M' , '2007-07-1' ,
'01545125910'),

(2865478925, 'mostafa' , 'tarek' , 'alahram' , 'M' , '2004-09-9' ,
'01157126303'),

(2965478925, 'ayaat' , 'tarek' , 'alahram' , 'F' , '1953-10-9' , '01057126404');

-- -------------

INSERT INTO manger
VALUES(9965412365),

(2111137605),

(4111113930),

(5111123090),

(2145678945),

(2245678978),
```

(6111137166),

(2345698796);

-- ------------

INSERT INTO station

VALUES('al-shohadaa' , 'Giza' , 5111123090),

('nasser' , 'Giza' , 2111137605),

('sadat' , 'Qalyubia' , 4111113930),

('Attaba' , 'Qalyubia' , 6111137166),

('cairoUN' , 'cairo' , 9965412365),

('ghamra' , 'Giza' , 2145678945),

('kitkat' , 'cairo' , 2245678978),

('shubra' , 'Qalyubia' , 2345698796);

-- ------------

INSERT INTO employee

VALUES(9965412365, 'cairoUN',3397,8),

(2111137605, 'nasser',2505,9),

(3111143747, 'cairoUN',8132,15),

(4111113930, 'sadat',4587,24),

(5111123090, 'al-shohadaa',4576,3),

(6111137166, 'Attaba',8811.123,7),

(7111125839, 'cairoUN',8719.99,12),

(8111141088, 'cairoUN',5055,13),

```sql
(9111142622, 'sadat',2200,4),

(1011137605, 'cairoUN',1005,8),

(1111139227, 'cairoUN',6666,9),

(1511134012, 'sadat',7894,6),

(1911142470, 'cairoUN',6000,3),

(2145678945, 'ghamra',8000,18),

(2345698796, 'shubra',4569,8),

(2245678978, 'kitkat',10000,24),

(2432178965, 'cairoUN',879.99,1),

(2512365478, 'Attaba',6547.5,9),

(2665478925, 'sadat',3652.099,8),

(2765478925, 'nasser',1212,4),

(2865478925, 'shubra',1000.5,3),

(2965478925, 'kitkat',300,10);
-- -------------
INSERT INTO Tr_pass
VALUES('25635KQ' , 'nasser', '11:56:22'),

('16278MG' , 'Attaba', '1:28:15'),

('19053KT' , 'Attaba', '2:36:34'),

('22394RR' , 'sadat', '21:41:13');
-- -------------
INSERT INTO safety_services
VALUES(1, 'cameras' , 'surveillance'),
```

(2, 'fire extinguisher' , 'Put out the fire'),

(3, 'alarm' , 'Alert'),

(4, 'metal-Detective' , 'Check');

-- -------------

INSERT INTO access_Method

VALUES(30925, 'ticket',10,'1', '2023-05-12' ,'0', null),

(2819, 'card',33,'0', null ,'1', 'student'),

(14728, 'card',55,0, null ,1, 'worker'),

(26494, 'ticket',10,1, '2022-09-25' ,0, null),

(24950, 'card',60,0, null ,1, 'older'),

(6189, 'ticket',10,1, '2023-01-06' ,0, null),

(31359, 'ticket',10,1, '2021-12-04' ,0, null),

(22392, 'card',38,0, null ,1, 'older'),

(20895, 'card',60,0, null ,1, 'adulte'),

(21087, 'ticket',5,1, '2020-05-05' ,0, null);

-- -------------

INSERT INTO passenger

VALUES(1211136745, '30925'),

(1311136061, '2819'),

(1411116284, '14728'),

(1611117300, '26494'),

(1711142801, '24950'),

```sql
(1811131612, '6189'),

(2011133463, '31359');

-- -------------

INSERT INTO buy

VALUES('31359',1211136745, 'green' ),

('26494',1411116284, 'red' ),

('21087',1311136061, 'yellow' ),

('30925',2011133463, 'green' );

-- -------------

INSERT INTO Ticket_Worker

VALUES(7111125839),

(8111141088),

(3111143747),

(9111142622);

-- -------------

INSERT INTO Subscription_Worker

VALUES(1011137605, 'nasser' ),

(1111139227, 'nasser' ),

(1511134012, 'cairoUN' ),

(1911142470, 'sadat' );

-- -------------

INSERT INTO T_has

VALUES('30925',7111125839),
```

```sql
('26494',8111141088),

('6189',3111143747);

-- ------------

INSERT INTO C_has

VALUES('2819',1011137605),

('14728',1111139227),

('22392',1511134012);

-- ------------

INSERT INTO Dependent

VALUES('ayaat','F','el-khalifa-el-maamoun','1920-03-04',2111137605),

('ashraf','M','mahmoud-el-attar','1964-08-28',3111143747),

('nade','F','el-mahdi','1968-05-01',1911142470);

-- ------------

INSERT INTO L_has

VALUES('ghamra' ,1),

('kitkat' ,3),

('shubra' ,2);

-- ------------

INSERT INTO transform

VALUES(2 ,3,'nasser'),

(2 ,1,'al-shohadaa'),

(1 ,2,'sadat');

-- ------------
```

INSERT INTO crosss

VALUES('20895','cairoUN','shubra'),

('24950','al-shohadaa','nasser'),

('21087','shubra','sadat'),

('30925','attaba','sadat');

-- -------------

INSERT INTO Accident

VALUES(1,'collision'),

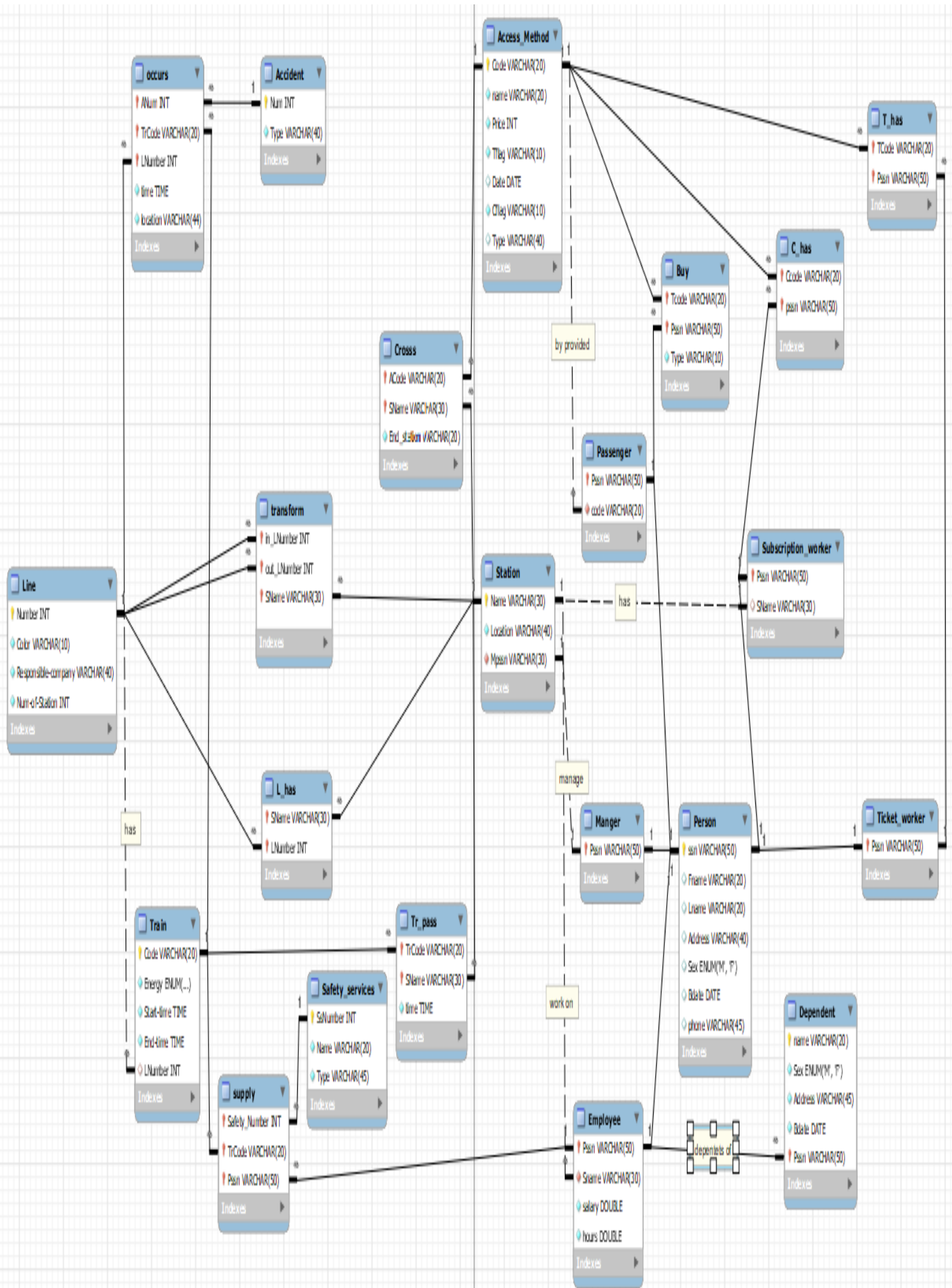(2,'firefighting'),

(3,'passengers die');

-- -------------

INSERT INTO occurs

VALUES(1, '16278MG', 2, '22:2:32', 'station'),

(2, '27531JI', 3, '18:15:49', 'train'),

(3, '19053KT', 1, '13:59:41', 'station');

-- -------------

INSERT INTO supply

VALUES(1,'19053KT',2432178965),

(2,'27531JI',2512365478),

(3,'16278MG',2345698796);

Workbench diagram :

Queries :

-- 1

-- retrieve all details about person.

SELECT *

FROM

person;

ssn	Fname	Lname	Address	Sex	Bdate	phone
1011137605	Ghanem	saber	mahmoud-el-attar	M	1994-10-14	01128669
1111139227	mazen	ibrahim	al-bashir	M	1966-05-04	011954
1211136745	ashraf	mohamed	el-mahdi	M	2002-03-24	01125734
1311136061	ibrahim	Ghanem	mahmoud-el-attar	M	1966-05-27	0113439
1411116284	ahmad	ibrahim	nour-el-deen	M	1977-12-23	01120942
1511134012	rahma	mohamed	el-mahdi	F	2007-07-11	01126339
1611117300	mazen	kareem	el-khalifa-el-maamoun	M	1983-09-18	01023464
1711142801	rahma	mazen	mahmoud-el-attar	F	1951-12-02	01121062
1811131612	mohamed	mazen	el-khalifa-el-maamoun	M	2007-12-16	01215058
1911142470	Ghanem	eldsoky	nour-el-deen	M	1979-11-15	01217726
2011133463	yasiin	said	el-mahdi	M	1983-12-16	01120636
2111137605	ibrahim	eldsoky	nour-el-deen	M	1970-02-27	01128218
2145678945	ibrahim	ashraf	abdelraouf	M	2004-12-16	01020636
2245678978	mohamed	ahmed	el-mahdi	M	2006-09-08	01093645
2345698796	ahmad	eldsoky	mogawra40	M	1907-03-30	0124485054
2432178965	adel	bola	mogawra45	M	1988-08-19	012554984
2512365478	sama	eldsoky	mogawra40	F	1906-09-10	01049464
2665478925	lolo	tarek	alahram	F	1907-07-09	01545125
2765478925	ali	mohamed	alahram	M	2007-07-01	01545125910
2865478925	mostafa	tarek	alahram	M	2004-09-09	01157126303
2965478925	ayaat	tarek	alahram	F	1953-10-09	01057126404
3111143747	shahd	ali	al-bashir	F	1948-07-22	0117634
4111113930	Gharib	mosa	el-khalifa-el-maamoun	M	1945-09-14	01129950
5111123090	ashraf	nader	nour-el-deen	M	2022-10-09	011387
6111137166	ashraf	rashad	el-khalifa-el-maamoun	M	1990-01-03	0117775
7111125839	taher	kmal	nour-el-deen	M	1993-02-11	01127710
8111141088	ashraf	mansor	el-mahdi	M	1953-02-02	01117895
9111142622	ayaat	ramy	mahmoud-el-attar	F	2009-01-22	0112564
9965412365	mohamed	eldsoky	el-mahdi	M	1942-04-23	01512814

-- 2 -----

   -- retrieve ticket code and date.

SELECT

   code, date

FROM

   access_method

WHERE

   date IS NOT NULL;

code	date
21087	2020-05-05
26494	2022-09-25
30925	2023-05-12
31359	2021-12-04
6189	2023-01-06

-- 3 -----

    -- retrieve train code and it's enargy that's in line 1.

```sql
SELECT
    code, energy
FROM
    train
WHERE
    LNumber = 1;
```

code	energy
10642LG	LOW
27531JI	FULL

-- 4 -----

-- retrieve station name where location is cairo ;

SELECT

   name

FROM

   station

WHERE

   Location = 'cairo';

name
cairoUN
kitkat

-- 5 -----

-- retrieve all details about station where station name contain the character'sh'.

```sql
SELECT
    *
FROM
    station
WHERE
    Name LIKE '%sh%';
```

Name	Location	Mpssn
al-shohadaa	Giza	5111123090
shubra	Qalyubia	2345698796
NULL	NULL	NULL

-- 6 -----

-- retrieve first name , ssn and address of all person that he's address contain characters 'mahdi'.

SELECT

    ssn, fname, address

FROM

    person

WHERE

    address LIKE '%mahdi%';

ssn	fname	address
1211136745	ashraf	el-mahdi
1511134012	rahma	el-mahdi
2011133463	yasiin	el-mahdi
2245678978	mohamed	el-mahdi
8111141088	ashraf	el-mahdi
9965412365	mohamed	el-mahdi
NULL	NULL	NULL

-- 7 -----

  -- retreve the name , ssn and phone of person who have WE number '015'

```sql
SELECT
   ssn, fname, phone
FROM
   person
WHERE
   phone LIKE '015%';
```

ssn	fname	phone
2665478925	lolo	01545125
2765478925	ali	01545125910
9965412365	mohamed	01512814
NULL	NULL	NULL

-- 8 -----

    -- retrieve the salary of empolyee that it contains a Fractions

SELECT

    salary

FROM

    employee

WHERE

    salary LIKE '%.%';

salary
879.99
6547.5
3652.099
1000.5
8811.123
8719.99

-- 9 -----

-- retrieve first name and number of persons that first character of his name is 'a' or secend character is 'a' for each name.

SELECT

    fname, COUNT(*) as "number of person"

FROM

    person

GROUP BY fname

having

    fname LIKE 'a%' OR fname LIKE '_a%';

fname	number of person
adel	1
ahmad	2
ali	1
ashraf	4
ayaat	2
mazen	2
rahma	2
sama	1
taher	1
yasiin	1

-- 10 -----

-- retrieve all salarys of empolyee that's not between 5000 and 8000.

SELECT

   salary

FROM

   employee

WHERE

   salary NOT BETWEEN 5000 AND 8000;

salary
1005
2505
10000
4569
879.99
3652.099
1212
1000.5
300
8132
4587
4576
8811.123
8719.99
2200
3397

-- 11 -----

-- retrieve first name , last name and salary of all empolyees whose salary between 5000 and 9000.

SELECT

    d.fname, d.lname, e.salary

FROM

    employee e,

    person d

WHERE

    salary BETWEEN 5000 AND 9000

       AND e.pssn = d.ssn;

fname	lname	salary
mazen	ibrahim	6666
rahma	mohamed	7894
Ghanem	eldsoky	6000
ibrahim	ashraf	8000
sama	eldsoky	6547.5
shahd	ali	8132
ashraf	rashad	8811.123
taher	kmal	8719.99
ashraf	mansor	5055

-- 12 -----

   -- retrieve first name and sex of ticket workers.

SELECT

   p.fname, p.sex

FROM

   person p

      JOIN

   ticket_worker t ON p.ssn = t.pssn

ORDER BY p.sex;

fname	sex
taher	M
ashraf	M
ayaat	F
shahd	F

-- 13 -----

-- retrivere all details of accident and show the time of each accident ordered desc.

SELECT

   *

FROM

   Accident d,

   occurs e

WHERE

   d.num = e.anum

ORDER BY time DESC;

Num	Type	ANum	TrCode	LNumber	time	location
1	collision	1	16278MG	2	22:02:32	station
2	firefighting	2	27531JI	3	18:15:49	train
3	passengers die	3	19053KT	1	13:59:41	station

-- 14 -----

-- retrieve all passenger that access-method is a card.

SELECT

  pssn , t.*

FROM

  Passenger p

    JOIN

  access_method T ON p.code = T.code

WHERE

  cflag = 1;

pssn	Code	name	Price	Tflag	Date	Cflag	Type
1411116284	14728	card	38	0	NULL	1	worker
1711142801	24950	card	38	0	NULL	1	older
1311136061	2819	card	38	0	NULL	1	student

-- 15 -----

-- retrivere all train that didn't make an accident.

SELECT

   d.code

FROM

   train d,

   occurs p

WHERE

   d.code <> p.trcode;

code
10642LG
10642LG
10642LG
27531JI
27531JI
16278MG
16278MG
22394RR
22394RR
22394RR
19053KT
19053KT
25635KQ
25635KQ
25635KQ

-- 16 -----

   -- retrieve all train that make an accident.

SELECT

   d.code

FROM

   train d,

   occurs p

WHERE

   d.code = p.trcode;

code
16278MG
27531JI
19053KT

-- 17 -----

-- retrieve the name , salary and salary after adding 10% interest of the empolyees.

SELECT

fname, lname, (0.1 * salary) + salary AS inc_sal, salary

FROM

person

JOIN

employee ON pssn = ssn;

fname	lname	inc_sal	salary
Ghanem	saber	1105.5	1005
mazen	ibrahim	7332.6	6666
rahma	mohamed	8683.4	7894
Ghanem	eldsoky	6600	6000
ibrahim	eldsoky	2755.5	2505
ibrahim	ashraf	8800	8000
mohamed	ahmed	11000	10000
ahmad	eldsoky	5025.9	4569
adel	bola	967.989	879.99
sama	eldsoky	7202.25	6547.5
lolo	tarek	4017.3089	3652.099
ali	mohamed	1333.2	1212
mostafa	tarek	1100.55	1000.5
ayaat	tarek	330	300
shahd	ali	8945.2	8132
Gharib	mosa	5045.7	4587
ashraf	nader	5033.6	4576
ashraf	rashad	9692.2353	8811.123
taher	kmal	9591.989	8719.99
ashraf	mansor	5560.5	5055
ayaat	ramy	2420	2200
mohamed	eldsoky	3736.7	3397

-- 18 -----

-- add column relatonship to dependent with type varchar.

alter table dependent

add relationship varchar(20) not null ;

-- 19 -----

-- change column relationship where name is ashraf such that it's relationship be son.

UPDATE dependent

SET

   relationship = 'son'

WHERE

  (name = 'ashraf')

    AND (Pssn = '3111143747');

-- 20 -----

 -- change column relationship where name is ayyat such that it's relationship be wife.

UPDATE dependent

SET

   relationship = 'wife'

WHERE

  (name = 'ayaat')

    AND (Pssn = '2111137605');

-- 21 -----

-- change column relationship where name is nade such that it's relationship be mother.

UPDATE dependent

SET

   relationship = 'mother'

WHERE

  (name = 'nade')

    AND (Pssn = '1911142470');

-- 22 -----

-- change the access method from ticket to card and it's price is 38 , where code of ticket is 6189.

```sql
UPDATE access_method
SET
    name = 'card',
    price = 38
WHERE
    code= 6189;
```

-- 23 -----

-- retrieve dependant name of employees.

```sql
SELECT
    e.name
FROM
    dependent e,
    employee d
WHERE
    d.pssn = e.pssn;
```

name
nade
ayaat
ashraf

-- 24 -----

-- retrieve first name , last name and salary grater than or equal 5000 from mangers.

SELECT

   p.fname, p.lname, e.salary

FROM

   person p

     JOIN

   employee e ON p.ssn = e.pssn

     JOIN

   manger m ON p.ssn = m.pssn

WHERE

   salary >= '5000';

fname	lname	salary
ibrahim	ashraf	8000
mohamed	ahmed	10000
ashraf	rashad	8811.123

-- 25 -----

-- retrieve station names and it's line number.

SELECT

   s.name, l.number

FROM

   station s,

   line l,

   l_has h

WHERE

   s.name = h.sname

      AND h.LNumber = l.Number;

name	number
ghamra	1
kitkat	3
shubra	2

-- 26 -----

-- retrieve the stations name that Connects two lines.

SELECT DISTINCT

   t.sname, t.in_lnumber 'in_line', t.out_lnumber 'out_line'

FROM

   station s,

   transform t,

   line l

WHERE

   s.name = t.sname

     AND l.number = t.in_lnumber

     OR l.number = t.out_lnumber;

sname	in_line	out_line
al-shohadaa	2	1
sadat	1	2
nasser	2	3

-- 27 -----

-- retrieve the station name and ssn of subscription_worker who work in this station.

SELECT

    pssn, p.fname, p.lname, name

FROM

    station s

        JOIN

    subscription_worker w ON name = sname

        JOIN

    person p ON ssn = pssn;

pssn	fname	lname	name
1511134012	rahma	mohamed	cairoUN
1011137605	Ghanem	saber	nasser
1111139227	mazen	ibrahim	nasser
1911142470	Ghanem	eldsoky	sadat

-- 28 -----

-- retrivere name , pssn , ticket code and date where the ticket type is green.

```sql
SELECT
    ps.pssn, p.fname, p.lname, b.tcode, date
FROM
    person AS p
        JOIN
    passenger AS ps ON p.ssn = ps.pssn
        JOIN
    buy AS b ON b.pssn = ps.pssn AND type = 'green'
        JOIN
    access_method AS c ON b.tcode = c.code;
```

pssn	fname	lname	tcode	date
2011133463	yasiin	said	30925	2023-05-12
1211136745	ashraf	mohamed	31359	2021-12-04

-- 29 ------

-- retrieve name and ssn for person that didnt work like manger , employee , ticet-worcker and subscription-worker.

```sql
SELECT
    e.pssn, p.fname
FROM
    person AS p
        JOIN
    employee AS e ON e.pssn = p.ssn
        LEFT outer JOIN
    manger AS m ON m.pssn = e.pssn
        LEFT outer JOIN
    ticket_worker AS t ON t.pssn = e.pssn
        LEFT outer JOIN
    subscription_worker AS s ON s.pssn = e.pssn
WHERE
    m.pssn IS NULL AND t.pssn IS NULL
        AND s.pssn IS NULL;
```

pssn	fname
2512365478	sama
2432178965	adel
2965478925	ayaat
2765478925	ali
2665478925	lolo
2865478925	mostafa

-- 30 -----

-- retrieve employee name that supply a safety services for the train.

```sql
SELECT
    fname
FROM
    employee AS e
        JOIN
    supply AS s ON e.pssn = s.pssn
        JOIN
    train AS t ON t.code = s.trcode
        JOIN
    safety_services AS f ON f.ssnumber = s.safety_number
        JOIN
    person p ON e.pssn = p.ssn;
```

fname
adel
sama
ahmad

-- 31 -----

-- retrieve the name of employee and station , train code , safety services types , time and number of an accident.

```sql
SELECT
    p.fname,
    p.lname,
    sname 'staition',
    t.code 'train code',
    v.type 'kind of safaty',
    o.time 'accident time',
    a.num 'accident number'
FROM
    person p
        JOIN
    employee e ON p.ssn = e.pssn
        JOIN
    supply s ON e.pssn = s.pssn
        JOIN
    train t ON s.trcode = t.code
```

JOIN

    occurs o ON t.code = o.trcode

    JOIN

    accident a ON o.ANum = a.num,

    safety_services v,

    station st

WHERE

    s.Safety_Number = v.SsNumber

    AND st.name = e.sname

ORDER BY a.num;

fname	lname	staition	train code	kind of safaty	accident time	accident number
ahmad	eldsoky	shubra	16278MG	Alert	22:02:32	1
sama	eldsoky	Attaba	27531JI	Put out the fire	18:15:49	2
adel	bola	cairoUN	19053KT	surveillance	13:59:41	3

-- 32 -----

-- retrieve the name of passengers, ticket type ,price ,code and date who went to sadat station.

```sql
SELECT
    p.fname,
    p.lname,
    b.type,
    a.price,
    c.acode 'ticket code',
    a.date
FROM
    person p
        JOIN
    passenger pa ON p.ssn = pa.pssn
        JOIN
    buy b ON pa.pssn = b.pssn
        JOIN
    access_method a ON b.tcode = a.code
        JOIN
```

```
    crosss c ON a.code = c.acode

        JOIN

    station s ON c.sname = s.name
WHERE

    c.End_station = 'sadat';
```

fname	lname	type	price	ticket code	date
ibrahim	Ghanem	yellow	5	21087	2020-05-05
yasiin	said	green	10	30925	2023-05-12

-- 33 -----

-- retrieve the name , birth date and the age ordered desc of the person.

SELECT

    fname , lname , bdate, TIMESTAMPDIFF(YEAR, bdate, CURDATE()) AS age

FROM

    person

ORDER BY 4 DESC;

fname	lname	bdate	age
sama	eldsoky	1906-09-10	117
ahmad	eldsoky	1907-03-30	116
lolo	tarek	1907-07-09	116
mohamed	eldsoky	1942-04-23	81
Gharib	mosa	1945-09-14	78
shahd	ali	1948-07-22	75
rahma	mazen	1951-12-02	72
ayaat	tarek	1953-10-09	70
ashraf	mansor	1953-02-02	70
mazen	ibrahim	1966-05-04	57
ibrahim	Ghanem	1966-05-27	57
ibrahim	eldsoky	1970-02-27	53
ahmad	ibrahim	1977-12-23	45
Ghanem	eldsoky	1979-11-15	44
yasiin	said	1983-12-16	40
mazen	kareem	1983-09-18	40
adel	bola	1988-08-19	35
ashraf	rashad	1990-01-03	33
taher	kmal	1993-02-11	30
Ghanem	saber	1994-10-14	29
ashraf	mohamed	2002-03-24	21
ibrahim	ashraf	2004-12-16	19
mostafa	tarek	2004-09-09	19
mohamed	ahmed	2006-09-08	17
ali	mohamed	2007-07-01	16
mohamed	mazen	2007-12-16	16
rahma	mohamed	2007-07-11	16
ayaat	ramy	2009-01-22	14
ashraf	nader	2022-10-09	1

-- 34 -----

-- retrieve sex and number for each dependant of employee.

SELECT

   sex, COUNT(*) "number of dependent"

FROM

   dependent

GROUP BY sex

having

   sex IN ('f' , 'm');

sex	number of dependent
M	1
F	2

-- 35 -----

-- retrieve the maximum and minimum of salary of employees.

SELECT

MAX(salary), MIN(salary)

FROM

employee;

MAX(salary)	MIN(salary)
10000	300

-- 36 -----

-- retrieve count of all salayrs that empolyees have , sumation of all salarys and average of salarys.

```
SELECT
    COUNT(salary) AS counts,
    SUM(salary) AS 'sum of salary',
    AVG(salary) AS "Average Salary"
FROM
    employee;
```

counts	sum of salary	Average Salary
22	105709.202	4804.963727272728

-- 37 -----

-- retrieve first name , ssn , last name and salary from employee where salary grater than average of salary of each name ordered.

```sql
SELECT
    e.pssn, p.fname, p.lname, e.salary
FROM
    person p JOIN employee e

    ON p.ssn = e.pssn
WHERE
    e.salary > (SELECT AVG(salary)

        FROM employee)

    ORDER BY fname;
```

pssn	fname	lname	salary
6111137166	ashraf	rashad	8811.123
8111141088	ashraf	mansor	5055
1911142470	Ghanem	eldsoky	6000
2145678945	ibrahim	ashraf	8000
1111139227	mazen	ibrahim	6666
2245678978	mohamed	ahmed	10000
1511134012	rahma	mohamed	7894
2512365478	sama	eldsoky	6547.5
3111143747	shahd	ali	8132
7111125839	taher	kmal	8719.99

-- 38 -----

-- retrieve the name and ssn of empolyees and manager where they work is staition nasser.

SELECT

ssn,fname, lname

FROM

employee AS e,

person AS p,

manger AS m,

station AS n

WHERE

e.pssn = p.ssn AND m.pssn = p.ssn

AND n.name = (SELECT name

FROM

station AS n

WHERE

n.name = 'nasser');

ssn	fname	lname
2111137605	ibrahim	eldsoky
2145678945	ibrahim	ashraf
2245678978	mohamed	ahmed
2345698796	ahmad	eldsoky
4111113930	Gharib	mosa
5111123090	ashraf	nader
6111137166	ashraf	rashad
9965412365	mohamed	eldsoky

-- 39 -----

-- retrieve all details from empolyee who works minimum of hours .

SELECT

  *

FROM

  Employee

WHERE

  hours = (SELECT

      MIN(hours)

    FROM

      Employee);

Pssn	Sname	salary	hours
2432178965	cairoUN	879.99	1
NULL	NULL	NULL	NULL

-- 40 -----

-- retrieve the names of employees who work with a specific manager.

```sql
SELECT
    fname, lname, ssn, salary, name
FROM
    person,
    employee,
    station
WHERE
    pssn = ssn AND sname = name
        AND mpssn IN (SELECT
            pssn
        FROM
            station,
            manger
        WHERE
            mpssn = pssn AND pssn = '4111113930');
```

fname	lname	ssn	salary	name
rahma	mohamed	1511134012	7894	sadat
lolo	tarek	2665478925	3652.099	sadat
Gharib	mosa	4111113930	4587	sadat
ayaat	ramy	9111142622	2200	sadat

ماذا اقول ..
واي قول يقال .. بعد الذي قيل

Thank you