

## Question 01 – MAIL\_ORDER

```
CREATE DATABASE mail_order;
```

```
USE mail_order;
```

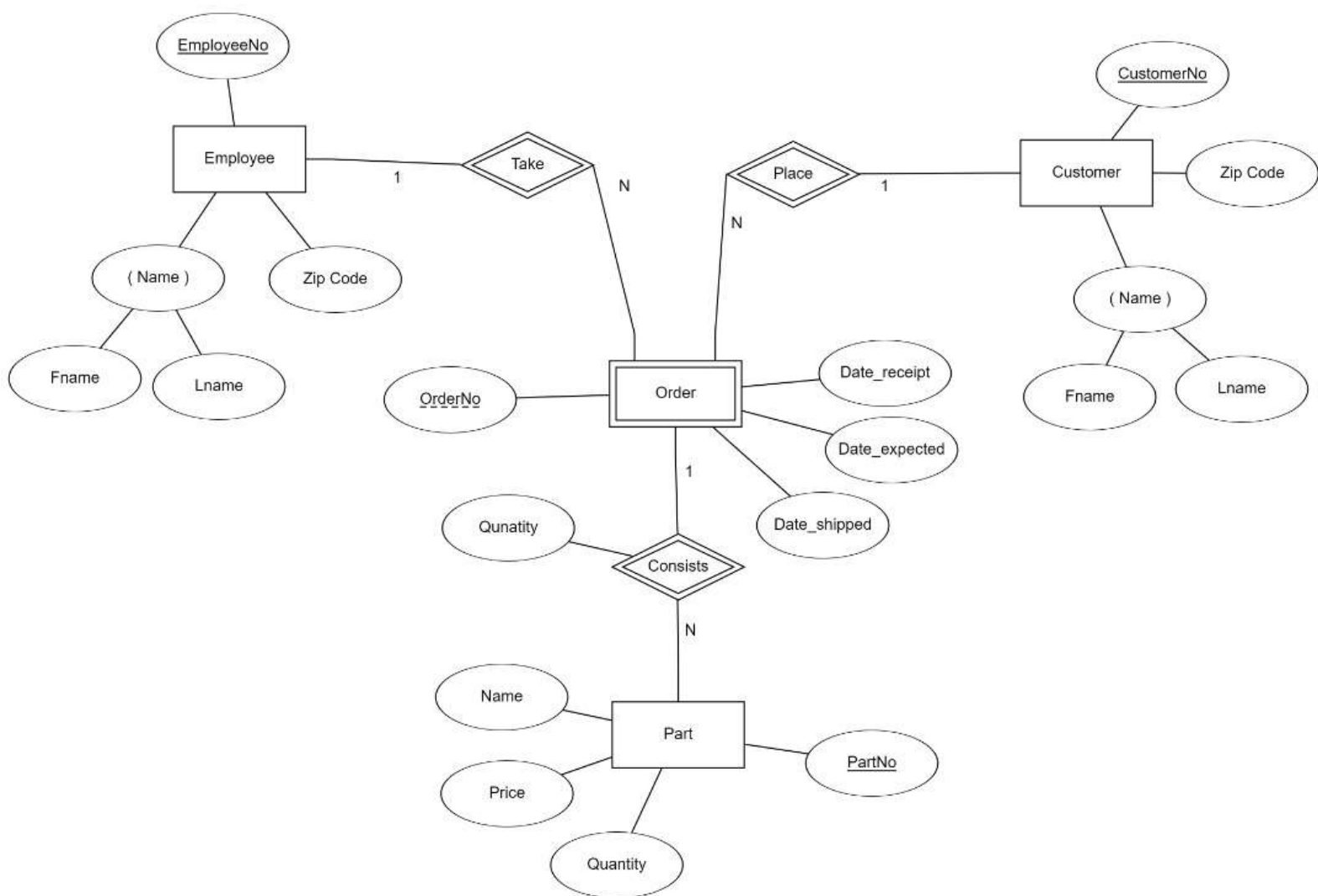
```
CREATE TABLE employee (  
    emp_no INT NOT NULL PRIMARY KEY,  
    first_name VARCHAR(20),  
    last_name VARCHAR (20),  
    zip_code CHAR (5) );
```

```
CREATE TABLE customer(  
    cust_no INT NOT NULL PRIMARY KEY,  
    first_name VARCHAR(20),  
    last_name VARCHAR (20),  
    zip_code CHAR (5) );
```

```
CREATE TABLE part(  
    part_no INT NOT NULL PRIMARY KEY,  
    part_name VARCHAR(20),  
    price INT,  
    quantity INT );
```

```
CREATE TABLE order(  
    order_no INT,  
    date_receipt Date,  
    date_expected Date,  
    date_shipped Date,  
    emp_no INT,
```

```
cust_no INT,  
part_no INT,  
PRIMARY KEY (order_no, emp_no, cust_no, part_no),  
FOREIGN KEY (emp_no) REFERENCES employee(emp_no),  
FOREIGN KEY (cust_no) REFERENCES customer(cust_no),  
FOREIGN KEY (part_no) REFERENCES part(part_no)  
);
```



## Question 02 – MOVIES

CREATE TABLE

```
actors(  
  actor_id TINYINT(3) NOT NULL UNIQUE AUTO_INCREMENT,  
  name VARCHAR(20) NOT NULL ,  
  date_of_birth DATE NOT NULL,  
  PRIMARY KEY (name, date_of_birth),  
  UNIQUE (name, date_of_birth)  
);
```

CREATE TABLE

```
directors(  
  director_id TINYINT(3) NOT NULL UNIQUE AUTO_INCREMENT,  
  name VARCHAR(20) NOT NULL,  
  date_of_birth DATE NOT NULL,  
  PRIMARY KEY (name, date_of_birth),  
  UNIQUE (name, date_of_birth)  
);
```

CREATE TABLE

```
production_company(  
  comp_id TINYINT(3) NOT NULL UNIQUE AUTO_INCREMENT,  
  name VARCHAR(20) NOT NULL,  
  address DATE NOT NULL,  
  PRIMARY KEY (name, address),  
  UNIQUE (name, address)  
);
```

CREATE TABLE

```
categories(  
    category_id TINYINT(3) NOT NULL UNIQUE AUTO_INCREMENT,  
    category VARCHAR(15),  
    CHECK (category IN ('horror', 'action', 'drama'))  
);
```

CREATE TABLE

```
movies(  
    movie_id TINYINT(3) NOT NULL UNIQUE AUTO_INCREMENT,  
    title VARCHAR(20) NOT NULL,  
    year_release YEAR NOT NULL,  
    PRIMARY KEY (title, year_release),  
    plot VARCHAR(100),  
    comp_id TINYINT(3),  
    FOREIGN KEY (comp_id) references production_company (comp_id )  
);
```

CREATE TABLE

```
quotes(  
    quote_id TINYINT(3) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    actor_id TINYINT(3),  
    movie_id TINYINT(3),  
    quote_text VARCHAR(100) NOT NULL,  
    FOREIGN KEY (actor_id) references actors (actor_id),
```

```
FOREIGN KEY (movie_id) references movies (movie_id)
);
```

```
CREATE TABLE
```

```
    roles(
        role_id TINYINT(3) NOT NULL AUTO_INCREMENT PRIMARY
KEY,
        role_name VARCHAR(20) NOT NULL,
        actor_id TINYINT(3),
        FOREIGN KEY (actor_id) references actors (actor_id),
        actor_if_director TINYINT(3) DEFAULT NULL,
        FOREIGN KEY (actor_if_director) references directors (director_id),
        movie_id TINYINT(3),
        FOREIGN KEY (movie_id) references movies (movie_id)
    );
```

```
CREATE TABLE
```

```
    movie_has_category(

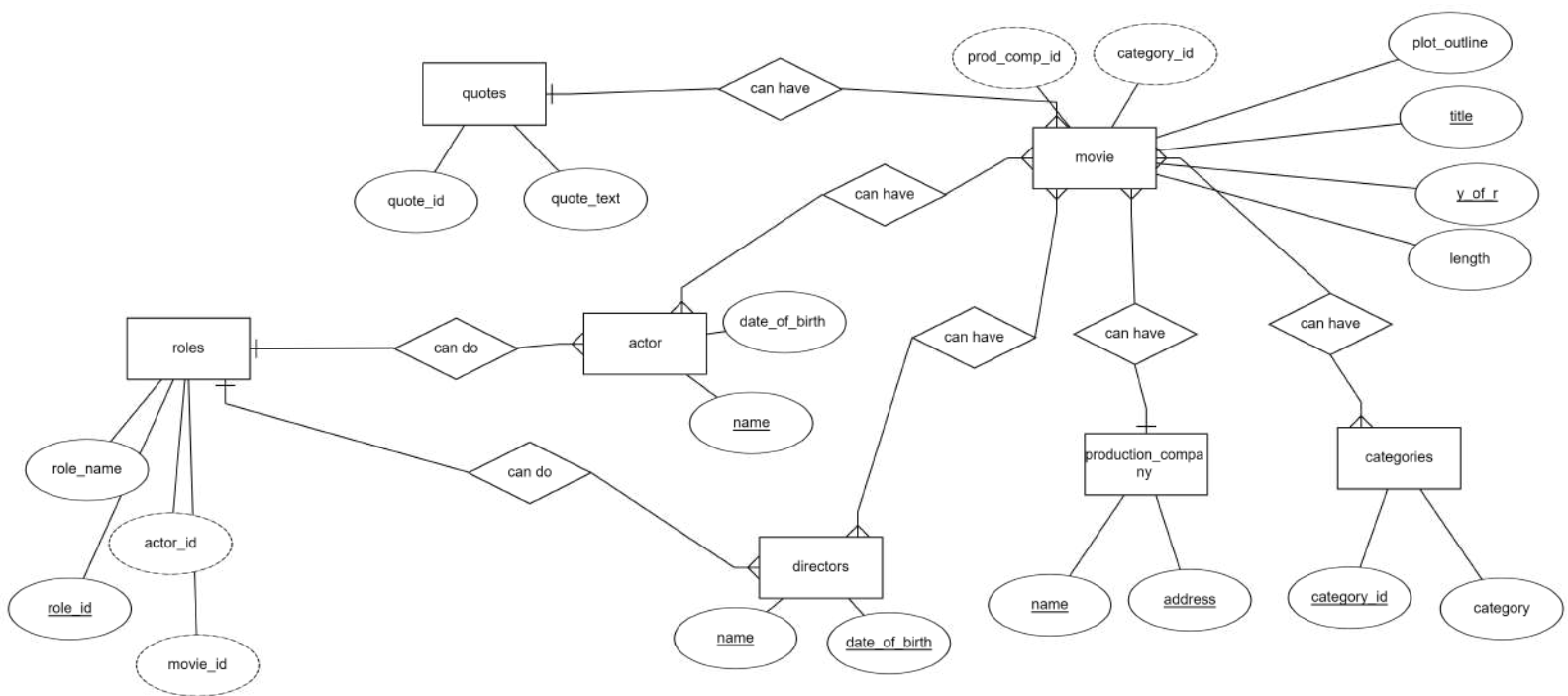
        category_id TINYINT(3),
        FOREIGN KEY (category_id) REFERENCES categories (category_id),
        movie_id TINYINT(3),
        FOREIGN KEY (movie_id) REFERENCES movies (movie_id)
    );
```

CREATE TABLE

```
    movie_has_directors(  
  
    director_id TINYINT(3),  
    movie_id TINYINT(3),  
    name VARCHAR(15),  
    FOREIGN KEY (director_id) REFERENCES directors (director_id),  
    FOREIGN KEY (movie_id) REFERENCES movies (movie_id),  
    FOREIGN KEY (name) REFERENCES directors (name)  
    );
```

CREATE TABLE

```
    movie_has_actors(  
  
    actor_id TINYINT(3),  
    movie_id TINYINT(3),  
    name VARCHAR(15),  
    FOREIGN KEY (actor_id) REFERENCES actors (actor_id),  
    FOREIGN KEY (movie_id) REFERENCES movies (movie_id),  
    FOREIGN KEY (name) REFERENCES actors (name)  
    );
```





### Question 03 – CONFERENCE\_REVIEW

```
CREATE TABLE emails(  
    email_id TINYINT(3) PRIMARY KEY AUTO_INCREMENT,  
    email VARCHAR(30)  
);  
  
CREATE TABLE authores(  
    email_id TINYINT(3) NOT NULL AUTO_INCREMENT,  
    UNIQUE(email_id),  
    FOREIGN KEY (`email_id`) REFERENCES `emails` (`email_id`) ON DELETE  
    RESTRICT ON UPDATE RESTRICT,  
    f_name VARCHAR(10),  
    l_name VARCHAR(10)  
);  
  
CREATE TABLE reviewers(  
    email_address VARCHAR(30) PRIMARY KEY,  
    f_name VARCHAR(10),  
    l_name VARCHAR(10),  
    affiliation VARCHAR(15),  
    topics_of_interests VARCHAR(30)  
);  
  
CREATE TABLE papers(  
    paper_id TINYINT(3) PRIMARY KEY AUTO_INCREMENT,  
    title VARCHAR(15),  
    abstract VARCHAR(100),  
    electronic_f VARCHAR(15)  
);
```

```

CREATE TABLE paper_reviews(
    review_id TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    reviewer_email VARCHAR(30) NOT NULL,
    FOREIGN KEY (`reviewer_email`) REFERENCES
`reviewers`(`email_address`) ON DELETE RESTRICT ON UPDATE RESTRICT,
    paper_id TINYINT(3) NOT NULL,
    FOREIGN KEY (`paper_id`) REFERENCES `papers`(`paper_id`) ON DELETE
RESTRICT ON UPDATE RESTRICT,
    rate_technical ENUM('1', '2', '3', '4', '5', '6', '7', '8', '9', '10'),
    rate_merit ENUM('1', '2', '3', '4', '5', '6', '7', '8', '9', '10'),
    rate_readability ENUM('1', '2', '3', '4', '5', '6', '7', '8', '9', '10'),
    rate_origanility ENUM('1', '2', '3', '4', '5', '6', '7', '8', '9', '10'),
    rate_relevance ENUM('1', '2', '3', '4', '5', '6', '7', '8', '9', '10'),
    comment_committe VARCHAR(100),
    feedback VARCHAR(100)
);

```

```

CREATE TABLE paper_has_reviewer(
    paper_id TINYINT(3),
    FOREIGN KEY (`paper_id`) REFERENCES `papers`(`paper_id`) ON DELETE
RESTRICT ON UPDATE RESTRICT,
    reviewer_id VARCHAR(30),
    FOREIGN KEY (`reviewer_id`) REFERENCES `reviewers`(`email_address`)
ON DELETE RESTRICT ON UPDATE RESTRICT
);

```

```

CREATE TABLE paper_has_author(
    paper_id TINYINT(3),

```

```
FOREIGN KEY (`paper_id`) REFERENCES `papers`(`paper_id`) ON DELETE  
RESTRICT ON UPDATE RESTRICT,
```

```
author_id TINYINT(3),
```

```
FOREIGN KEY (`author_id`) REFERENCES `authors`(`email_id`) ON  
DELETE RESTRICT ON UPDATE RESTRICT,
```

```
contact_author ENUM ('yes', 'no') NOT NULL
```

```
);
```

