Khoi Duong

Prof. Yang

CS457

11/5/2022

HW#4A

Exercise 6.5

The primary keys will be underlined and foreign keys will be bolded. If a key is both primary and foreign key, it will be bolded and underlined.

STUDENT(Name, Student_number, Class, Major)

COURSE(Course_name, Course_number, Credit_hours, Department)

PREREQUISITE(**Course_number**, **Prerequisite_number**)

SECTION(Section_identifier, Course_number, Semester, Year, Instructor)

GRADE_REPORT(**Student_number**, **Section_identifier**, Grade)

SQL DDL statements to define the database

```
CREATE TABLE STUDENT
(
Name VARCHAR(20) NOT NULL,
Student_number INT PRIMARY KEY,
Class VARCHAR(20),
Major VARCHAR(10)
```

```
);

CREATE TABLE COURSE
(
Course_name VARCHAR(20) NOT NULL,
Course_number VARCHAR(10) PRIMARY KEY,
Credit_hours DOUBLE,
Department VARCHAR(10)
);

CREATE TABLE PREREQUISITE
(
Course_number VARCHAR(10) REFERENCES COURSE(Course_number),
Prerequisite_number INT REFERENcES COURSE(Course_number),
PRIMARY KEY(Course_number,Prerequisite_number)
);

CREATE TABLE SECTION
(
Section_identifier INT PRIMARY KEY,
Course_number VARCHAR(10) REFERENCES COURSE(Course_number),
Semester VARCHAR(10),
Year INT,
Instructor VARCHAR(20)
);


CREATE TABLE GRADE_REPORT
(
Student_number INT REFERENCES STUDENT(Studnet_number),
Section_identifer INT REFERENCES SECTION(Section_identifer),
Grade CHAR(1),
PRIMARY KEY(Student_number,Section_identifer)
);
```

Exercise 6.7

The primary keys will be underlined and foreign keys will be bolded. If a key is both primary and foreign key, it will be bolded and underlined.

AIRPORT(Aiport_code, Name, City, State)

FLIGHT(Flight_number, Airline, Weekdays)

FLIGHT_LEG(**Flight_number**, Leg_number, Departure_airport_code, Scheduled_departure_time, Arrival_airport_code, Scheduled_arrival_time)

LEG_INSTANCE(**Flight_number**, **Leg_number**, Date, Number_of_available_seats, Airplane_id, Departure_airport_code, Departure_time, Arrival_airport_code, Arrival_time)

FARE(**Flight_number**, Fare_code, Amount, Restrictions)

AIRPLANE_TYPE(Airplane_type_name, Max_seats, Company)

CAN_LAND(**Airplane_type_name**, **Airport_code**)

AIRPLANE(Airplane_id, Total_number_of_seats, Airplane_type)

SEAT_RESERVATION(**Flight_number**, **Leg_number**, Date, Seat_number, Customer_name, Customer_phone)

SQL DDL statements to define the database

AIRPORT:

CREATE TABLE AIRPORT (AIRPORT_CODE CHAR (3) NOT NULL, NAME VARCHAR (30) NOT NULL, CITY VARCHAR (30) NOT NULL, STATE VARCHAR (30), PRIMARY KEY (AIRPORT_CODE));

FLIGHT:

CREATE TABLE FLIGHT (NUMBER VARCHAR (6) NOT NULL, AIRLINE VARCHAR (20) NOT NULL, WEEKDAYS VARCHAR (10) NOT NULL, PRIMARY KEY (NUMBER));

FLIGHT_LEG:

CREATE TABLE FLIGHT_LEG (FLIGHT_NUMBER VARCHAR (6) NOT NULL,
LEG_NUMBER INTEGER NOT NULL, DEPARTURE_AIRPORT_CODE CHAR (3) NOT NULL, SCHEDULED_DEPARTURE_TIME TIMESTAMP WITH TIME ZONE,
ARRIVAL_AIRPORT_CODE CHAR (3) NOT NULL, SCHEDULED_ARRIVAL_TIME TIMESTAMP WITH TIME ZONE, PRIMARY KEY (FLIGHT_NUMBER, LEG_NUMBER), FOREIGN KEY (FLIGHT_NUMBER) REFERENCES FLIGHT (NUMBER), FOREIGN KEY (DEPARTURE_AIRPORT_CODE) REFERENCES AIRPORT (AIRPORT_CODE), FOREIGN KEY (ARRIVAL_AIRPORT_CODE) (AIRPORT_CODE));

LEG_INSTANCE:

CREATE TABLE LEG_INSTANCE (FLIGHT_NUMBER VARCHAR (6) NOT NULL, LEG_NUMBER INTEGER NOT NULL, LEG_DATE DATE NOT NULL, NO_OF_AVAILABLE_SEATS INTEGER, AIRPLANE_ID INTEGER,

DEPARTURE_AIRPORT_CODE CHAR(3), DEPARTURE_TIME TIMESTAMP WITH TIME ZONE, ARRIVAL_AIRPORT_CODE CHAR(3), ARRIVAL_TIME TIMESTAMP WITH TIME ZONE, PRIMARY KEY (FLIGHT_NUMBER, LEG_NUMBER, LEG_DATE), FOREIGN KEY (FLIGHT_NUMBER, LEG_NUMBER) REFERENCES FLIGHT_LEG (FLIGHT_NUMBER, LEG_NUMBER), FOREIGN KEY (AIRPLANE_ID) REFERENCES

AIRPLANE (AIRPLANE_ID), FOREIGN KEY (DEPARTURE_AIRPORT_CODE) (AIRPORT_CODE),

FOREIGN KEY (ARRIVAL_AIRPORT_CODE) (AIRPORT_CODE));

FARES:

CREATE TABLE FARES (FLIGHT_NUMBER VARCHAR (6) NOT NULL,

FARE_CODE VARCHAR (10) NOT NULL, AMOUNT DECIMAL (8, 2) NOT NULL,

RESTRICTIONS VARCHAR (200), PRIMARY KEY (FLIGHT_NUMBER, FARE_CODE), FOREIGN KEY (FLIGHT_NUMBER) REFERENCES FLIGHT (NUMBER));

AIRPLANE_TYPE:

```
CREATE TABLE AIRPLANE_TYPE (TYPE_NAME VARCHAR (20) NOT NULL,
MAX_SEATS INTEGER NOT NULL, COMPANY VARCHAR (15) NOT NULL,
PRIMARY KEY (TYPE_NAME) );
```

CAN_LAND:

```
CREATE TABLE CAN_LAND (AIRPLANE_TYPE_NAME VARCHAR (20) NOT NULL,
AIRPORT_CODE CHAR (3) NOT NULL, PRIMARY KEY (AIRPLANE_TYPE_NAME,
AIRPORT_CODE), FOREIGN KEY (AIRPLANE_TYPE_NAME) REFERENCES
AIRPLANE_TYPE (TYPE_NAME),
FOREIGN KEY (AIRPORT_CODE) (AIRPORT_CODE) );
```

AIRPLANE:

```
CREATE TABLE AIRPLANE (AIRPLANE_ID INTEGER NOT NULL,
TOTAL_NUMBER_OF_SEATS INTEGER NOT NULL, AIRPLANE_TYPE VARCHAR (20)
NOT NULL, PRIMARY KEY (AIRPLANE_ID),
FOREIGN KEY (AIRPLANE_TYPE) REFERENCES AIRPLANE_TYPE (TYPE_NAME) );
```

SEAT_RESERVATION:

```
CREATE TABLE SEAT_RESERVATION (FLIGHT_NUMBER VARCHAR (6) NOT NULL,
LEG_NUMBER INTEGER NOT NULL, LEG_DATE DATE NOT NULL,
SEAT_NUMBER VARCHAR (4), CUSTOMER_NAME VARCHAR (30) NOT NULL,
CUSTOMER_PHONE CHAR (12), PRIMARY KEY (FLIGHT_NUMBER, LEG_NUMBER,
LEG_DATE, SEAT_NUMBER), FOREIGN KEY (FLIGHT_NUMBER, LEG_NUMBER,
LEG_DATE) REFERENCES
LEG_INSTANCE (FLIGHT_NUMBER, LEG_NUMBER, LEG_DATE) );
```

Exercise 6.8

The primary keys will be underlined and foreign keys will be bolded. If a key is both primary and foreign key, it will be bolded and underlined.

BOOK(Book_id, Title, Publisher_name)

BOOK_AUTHORS(**Book_id**, Author_name)

PUBLISHER(Name, Address, Phone)

BOOK_COPIES(**Book_id**, Branch_id, No_of_copies)

BOOK_LOANS(**Book_id**, **Branch_id**, Card_no, Date_out, Due_date)

LIBRARY_BRANCH(**Branch_id**, Branch_name, Address)

BORROWER(**Card_no**, Name, Address, Phone)

SQL DDL statements to define the database

```
CREATE TABLE Book (

Book_id Int PRIMARY KEY,

Title Varchar(200),

Publisher_name Varchar(200),
FOREIGN KEY (Publisher_name) REFERENCES Publisher(Name)ON DELETE SET NULL
ON UPDATE CASCADE
);

CREATE TABLE Book_Authors (
Book_id Int NOT NULL,
Author_name Varchar(200) NOT NULL,
PRIMARY KEY (Book_id, Author_name),
FOREIGN KEY (Book_id) REFERENCES Book(Book_id)
ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Publisher (
Name Varchar(200) PRIMARY KEY,
Address Varchar(400),
Phone Decimal(20)
);

CREATE TABLE Book_Copies (
Book_id Int NOT NULL,
Branch_id Char(4) NOT NULL,
No_of_copies Int DEFAULT 1,
PRIMARY KEY (Book_id, Branch_id),
FOREIGN KEY (Book_id) REFERENCES Book(Book_id)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (Branch_id) REFERENCES Library_Branch(Branch_id)
ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Book_Loans (
Book_id Int NOT NULL,
Branch_id Char(4) NOT NULL,
Card_no Int NOT NULL,
Date_out Date,
Due_date Date,
PRIMARY KEY (Book_id, Branch_id, Card_no),
FOREIGN KEY (Book_id) REFERENCES Book(Book_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
FOREIGN KEY (Branch_id) REFERENCES Library_Branch(Branch_id)
ON DELETE RESTRICT ON UPDATE CASCADE,
FOREIGN KEY (Card_no) REFERENCES Borrower(Card_no)
ON DELETE RESTRICT ON UPDATE CASCADE
);

CREATE TABLE Library_Branch (
Branch_id Char(4) PRIMARY KEY,
Branch_name Varchar(200) NOT NULL,
Address Varchar(400)
);

CREATE TABLE Borrower (
Card_no Int PRIMARY KEY,
Name Varchar(200) NOT NULL,
Address Varchar(400),
Phone Decimal(20)
);
```