CHAPTER 3

3.1 Define the following terms as they apply to the relational model of data:
domain, attribute, n-tuple, relation schema, relation state, degree of a relation, relational database schema, and relational database state.

SOLUTION

Domain
It is a set of atomic (indivisible) values that can appear in a particular column in a relational schema. A common method of specifying domain is to specify a data type (integer, character, floating point, etc...) from which the data values forming a domain can be drawn

Relational Schema : R1 --> R(A1, A2, …..AN)
Attributes --> A1, A2 … AN
Domain of A1 ---> dom(A1)
Tuple --> t

Attribute
An attribute is a role played by some domain in the relational schema
In relational schema STUDENT, NAME can be one of the attributes of the relation.

n-tuple
If a Relational Schema consists of n Attributes, i.e., degree of relational schema is n, then n-tuple is an ordered list of n values that represent a tuple , t = ; where each value vi,1<=i<=n, is an element of dom(Ai) or is a special NULL value.

Relational Schema
It is a collection of attributes that define facts and relation between a real world entity and name.
Relational Schema : R1 --> R(A1, A2, …..AN)
Attributes --> A1, A2 … AN

Relation State
A relation state, r, of a relation schema R(A1, A2,……An), is a set of n-tuples. In another words a relation state of a relational schema is a collection of various tuples, where each tuple represents information about single entity.

Degree of a Relation
The degree of a relation is the number of attributes n of its relational schema

Relational database schema
A Relational Database Schema S is a set of relation schemas, S = { R1,R2,….Rn} and a set of integrity constraints IC.

Relational database state
A Relational Database State DB of S is set of relation states, DB = {r1,r2,….rn}, such that each ri is state of Ri and such that the ri relation states satisfy the integrity constraints specified in IC.

3.3 Why are duplicate tuples not allowed in a relation?

SOLUTION

Duplicate tuples are not allowed in a relation as it violates the relational integrity constraints

1. A key constraint states that there must be an attribute or combination of attributes in a relation whose values are unique
2. There should not be any two tuples in a relation whose values are same for their attribute values
3. If the tuples contain duplicate values, then it violates the key constraint

Hence duplicate tuples are not allowed in a relation

3.4 What is the difference between a key and a superkey?

SOLUTION

Key	Super Key
Key is a minimal super key.	A super key SK is a set of attributes that uniquely identifies the tuples of a relation.
A key cannot have redundant attributes.	A super key can have redundant attributes.
A key is a super key.	A super key is not a key.

3.8 Discuss the entity integrity and referential integrity constraints. Why is each considered important?

SOLUTION

Entity Integrity Constraint
It states that no primary key value can be NULL

IMPORTANCE :
Primary keys are used to identify a tuple in a relation.
Having a NULL value for primary key will mean that we cannot identify some tuples

Referential Integrity Constraint
It states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation

IMPORTANCE:
Referential Integrity Constraints are specified among two relations and are used to maintain consistency among tuples in two relations

3.9 Define foreign key. What is this concept used for?

SOLUTION

A foreign key is an attribute or composite attribute of one relation which is/are a primary key of other relation that is used to maintain relationship between two relations.

1. A relation can have more than one foreign key
2. A foreign key can contain null values

The concept of foreign key is used to maintain referential integrity constraint between two relations and hence in maintaining consistency among tuples in two relations.

3.11 Suppose that each of the following Update operations is applied directly to the database state shown in Figure 3.6. Discuss all integrity constraints violated by each operation, if any, and the different ways of enforcing these constraints.

a,d,f,h,I

SOLUTION

Insert <'Robert', 'F', 'Scott', '943775543', '1972-06-21', '2365 Newcastle Rd, Bellaire, TX', M, 58000, '888665555', 1> into EMPLOYEE.

No constraint is violated.

Insert <'677678989', NULL, '40.0'> into WORKS_ON.

Not acceptable.

1. Violates entity integrity constraint and referential integrity constraint.
2. Value of one of the attributes of primary is NULL

Enforcing :
1. Not performing operation and explain to user
2. Prompting user to specify correct values for the primary key and performing the operation.

Delete the WORKS_ON tuples with Essn = '333445555'.

No constraint is violated

Delete the PROJECT tuple with Pname = 'ProductX'.

Violates referential integrity constraint as value of Pnumber has been used as foreign key of WORKS_ON.

Modify the Mgr_ssn and Mgr_start_date of the DEPARTMENT tuple with Dnumber = 5 to '123456789' and '2007-10-01', respectively.

No constraint is violated

3.12 Consider the AIRLINE relational database schema shown in Figure 3.8, which describes a database for airline flight information. Each FLIGHT is identified by a Flight_number, and consists of one or more FLIGHT_LEGs with Leg_numbers 1, 2, 3, and so on. Each FLIGHT_LEG has scheduled arrival and departure times, airports, and one or more LEG_INSTANCEs—one for each Date on which the flight travels. FAREs are kept for each FLIGHT. For each FLIGHT_LEG instance, SEAT_RESERVATIONs are kept, as are the AIRPLANE used on the leg and the actual arrival and departure times and airports. An AIRPLANE is identified by an Airplane_id and is of a particular AIRPLANE_TYPE. CAN_LAND relates AIRPLANE_TYPEs to the AIRPORTs at which they can land. An AIRPORT is identified by an Airport_code. Consider an update for the AIRLINE database to enter a reservation on a particular flight or flight leg on a given date.

SOLUTION

a. It is necessary to check if the seats are available on a particular flight or flight leg

Check LEG_INSTANCE relation.
SELECT Number_of_available_seats FROM LEG_INSTANCE
WHERE Flight_Number = 'FL01' and Date = '2021-03-03'
If the Number_of_available_seats > 0, then perform the following
INSERT INTO SEAT_RESERVATION VALUES ('FL01', '1', '2021-03-03', '1', 'John', '9910110110')

b. Constraints
    a. Check if Number_of_available_seats in LEG_INSTANCE relation for the particular flight on particular date is greater than 1
    b. Check if the particular SEAT_NUMBER for a particular flight on the particular date is available or not

c.
    a. Checking the Number_of_available_seats in LEG_INSTANCE relation does not come under entity or referential integrity constraint.
    b. Checking for SEAT_NUMBER particular flight on the particular date comes under entity integrity constraint.

d. The referential integrity constraints hold are as follows:
    a. Flight_number of FLIGHT_LEG relation is a foreign key which references the Flight_number of FLIGHT relation.
    b. Flight_number of LEG_INSTANCE is a foreign key which references the Flight_number of FLIGHT relation.
    c. Flight_number of FARE is a foreign key which references the Flight_number of FLIGHT relation.
    d. Flight_number of SEAT_RESERVATION is a foreign key which references the Flight_number of FLIGHT relation.
    e. Departure_airport_code and Arrival_airport_code of FLIGHT_LEG are foreign keys which references the Airport_code of AIRPORT relation.
    f. Departure_airport_code and Arrival_airport_code of LEG_INSTANCE are foreign keys which references the Airport_code of AIRPORT relation.
    g. Airport_code of CAN_LAND is a foreign key which references the Airport_code of AIRPORT relation.
    h. Flight_number and Leg_number of LEG_INSTANCE are foreign keys which references Flight_number and Leg_number of FLIGHT_LEG.

3.17

Consider the following relations for a database that keeps track of automobile sales in a car dealership (OPTION refers to some optional equipment installed on an automobile):

CAR(Serial_no, Model, Manufacturer, Price)
OPTION(Serial_no, Option_name, Price)
SALE(Salesperson_id, Serial_no, Date, Sale_price)
SALESPERSON(Salesperson_id, Name, Phone)

First, specify the foreign keys for this schema, stating any assumptions you make.Next, populate the relations with a few sample tuples, and then give an example of an insertion in the SALE and SALESPERSON relations that violates the referential integrity constraints and of another insertion that does not.

<u>SOLUTION</u>

Foreign key constraints are as follows

a. Serial_no from OPTION is FK for CAR: spare parts can be added to cars with serial number
b. Serial_no from is SALE FK for CAR: only car with serial number can be put to sale
c. Salesperson_id from is FK for SALESPERSON : salesperson can sell any car

CHAPTER 9

Figure 9.8 shows an ER schema for a database that can be used to keep track of transport ships and their locations for maritime authorities. Map this schema into a relational schema and specify all primary keys and foreign keys.

SOLUTION

Following are the steps to convert the given ER scheme into a relational schema:
Step 1: Mapping the regular entity types:
Identify the regular entities in the given ER scheme and create a relation for each regular entity. Include all the simple attributes of regular entities into relations.
The relations are SHIP, SHIP_TYPE, STATE_COUNTRY, and SEA/OCEAN/LAKE.

SHIP

<u>Sname</u>	Owner	Ship_type	Pname

SHIP_TYPE

<u>Type</u>	Tonnage	Hull

STATE_COUNTRY

<u>Name</u>	Continent

SEA/OCEAN/LAKE

<u>SeaName</u>

Step 2: Mapping the weak entity types:
The weak entities in the given ER scheme are SHIP_MOVEMENT, PORT, and PORT_VISIT.
Create a relation for each weak entity. Include all the simple attributes of weak entities into relations and include the primary key of the strong entity that corresponds to the owner entity type as a foreign key.

SHIP_MOVEMENT

<u>Statename</u>	<u>Date</u>	<u>Time</u>	Latitude	Longitude

PORT

<u>Pname</u>	Seaname

PORT_VISIT

<u>VSname</u>	Start_date	End_date

Step 3: Mapping of binary 1:1 relationship types:
There exists one binary 1:1 relationship mapping which is SHIP_AT_PORT in given ER scheme.

Step 4: Mapping of binary 1: N relationship types:
1: N relationship types in given ER scheme are HISTORY, TYPE, IN, ON, HOME_PORT.
For HISTOR...handled in step 2.
For TYPE 1:N relationship type, include the primary key of SHIP_TYPE in SHIP.

SHIP

Sname	Owner	Ship_type

For IN 1: N relationship type, include the primary key of STATE_COUNTRY in PORT.
For ON 1: N relationship type, include the primary key of SEA/OCEAN/LAKE in PORT.

For HOME_PORT 1:N relationship type, include the primary key of PORT_VISIT in SHIP.

PORT

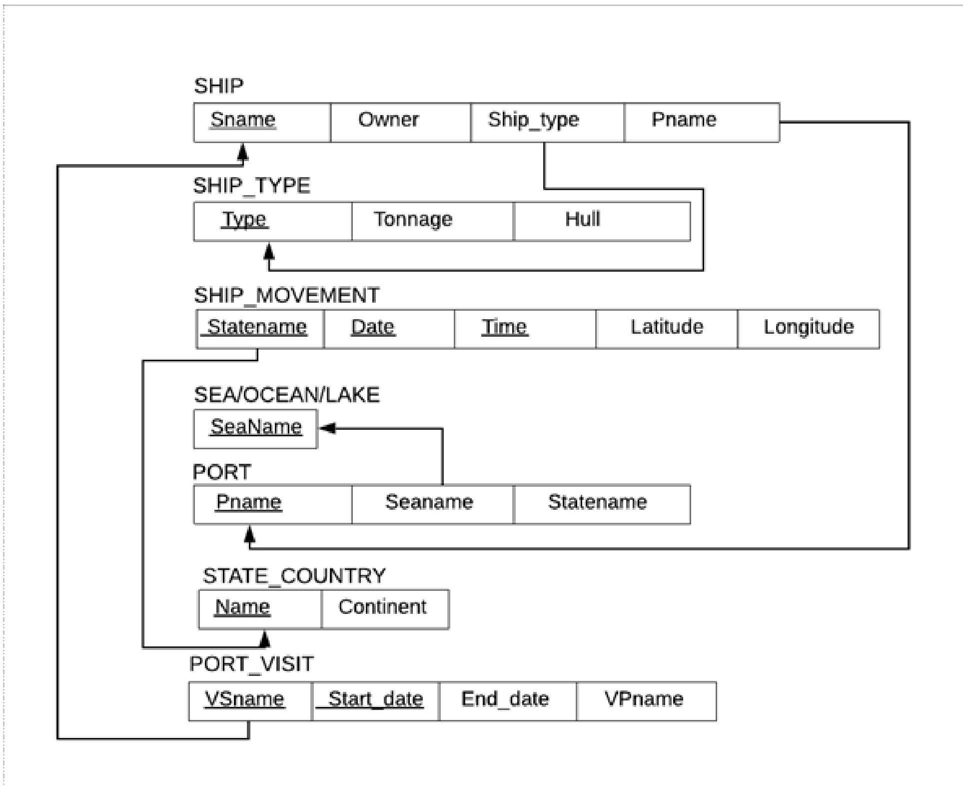
Pname	Seaname	Statename

SHIP

Sname	Owner	Ship_type	Pname

Step 5: Mapping of binary M: N relationship types:
There are no binary M: N relationship types in the given ER scheme.

Step 6: Mapping of multivalued attributes:
There are no multivalued attributes in the given ER scheme.
The relational schema is shown below:



The primary keys in the schema are:
SHIP: SnameSHIP_TYPE: TypeSHIP_MOVEMENT: Statename, Date, Time (Compound key)
SEA/OCEAN/LAKE: SeaNamePORT: PnameSTATE_COUNTRY: NamePORT_VISIT: VSname, Start_date
(Compound Key)

The foreign keys in the schema are:
SHIP: Ship_type, P_nameSHIP_TYPE: NoneSHIP_MOVEMENT: StatenameSEA/OCEAN/LAKE:
NonePORT: NoneSTATE_COUNTRY: NamePORT_VISIT: VSname

EXTRA EXERCISES :

a) STEP-1 : Mapping the regular entity types

PLANE_TYPE	Model	Capacity	Weight

AIRPLANE	Reg #

HANGAR	Number	Capacity	Location