3.1

**Domain** is a set of atomic values that can appear in a particular column in a relational schema. A common method of specifying domain is to specify a data type (such as int, char, float, etc.) from which the data values forming a domain can be drawn.

An **Attribute** is a role played by some domain in the relational schema.

**N-tuple** is an ordered list of n values that represent a tuple where each value is an element of domain or is a special NULL value.

**Relational Schema** is the collection of attributes that define facts and relation between a real world entity and name. In other words, a relational schema R, denoted by R (A1, A2, …, An) is made up of a name and a list of attributes A1, A2, …, An.

**A relation state** R of a relation schema R(A1, A2, …, An) is a set of n-tuples. It is a collection of various tuples where each tuple represents info of single entity. The formal definition is: a relation state R is a mathematical relation of degree n on the domains of all attributes, which is a subset of the cartesian product of the domains that define R.

The **degree of a relation** is the number of attributes n of its relational schema.

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

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

3.3

Because it violates the relational integrity constraints:

* A key constraint states that there must be an attribute or combination of attributes in a relation whose values are unique.
* There should not be any two tuples in a relation whose values are same for their attribute values.

If the tuples contain duplicate values, then it violates the key constraint. Thus, duplicates are not allowed in a relation.

3.4

**A Key must be a subset of the superkey.** As mentioned in the previous question, duplicates are not allowed in a relation. Then a relation must have identical values. This is called uniqueness constraint.

A superkey specifies a uniqueness constraint, but a key K of a relation schema R is a superkey of R with the additional property that removing any attribute A from K leaves a set of attributes K’ that is not a superkey of R any more. A key is a minimal superkey.

3.8

**Entity Integrity Constraint** states that no primary key value can be NULL. It is **important** because the primary key values are used to identify a tuple in a relation. Having Null value for primary key will mean that we cannot identify some tuples.

**Referential Integrity Constraints** states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation. Foreign Key is introduced (in the next question) to solve the referring relations. RIC is **important** since it is specified among two relations and are used to maintain consistency among tuples in two relations.

3.9

For two relational schemas R1 and R2, a set of attributes FK in relation schema R1 is foreign key of R1 that references relation R2 if it satisfies the following conditions:

* Attributes in FK have some domains as primary key attributes PK of R2; the attributes FK are said to reference relation R2.
* A value of FK in a tuple t1 of the current state r1(R1) either occurs in as a value of PK for some tuple in the current state r2 (R2) or is NULL. In former case tuple t1 is said to refer to the tuple t2.

When these two conditions hold true between R1 the referencing relation and R2 the referenced relation the referential integrity constraint is said to hold true.

This concept is used to specify integrity constraint between two relations and hence in maintaining consistency among tuples in two relations.

3.11(b, c, e, f, h)

B. Violates the referential integrity constraint. The Dnum is a foreign key referring to the Dnumber in the Department relation. There is no such thing as #2. Ways of enforcing: Reject the operation and warning. Or allowing the operation but pushing the user to input the instants of department #2. Or inserting NULL value in the Dnum so that this instant will not involving in the relationship.

C. Violates the Key constraint (Uniqueness constraint). Department with Dnumber 4 is already exist. Way of enforcing: Reject the operation and throw a warning to the user.

E. Acceptable.

F. Acceptable, both instants will be deleted.

H. Violates the referential integrity constraint as value Pnumber is referred by Pno in the works\_on relation. Ways of enforcing: Reject the operation and warning the user. Or delete all the records in different tables involving in (referring to) the “ProductX”.

3.12

A. Let flight number -FL1, Leg number -1, date –2017-09-21, SeatNumber allocated -20, Customer name -Smith and phone number -1011011001.

Insert < ‘FL1’, ‘1’, ‘2017-09-21’, ‘20’, ‘Smith’, ‘1011011001> into SEAT\_RESERVATION.

B. We need to check the following constraints:

Asked flight number of flight leg is available on given data. Data can be checked from LEG-INSTANCE table.

A non-reserved seat must exist for specifies date and flight. We can get total number of seats available from AIRPLANE.

C.

Key: Filght\_number + Leg\_number + Date + Seat\_number.

Entity Integrity constraint: A non-reserved seat must exist for specifies date and flight and leg-number so that there is a unique entity in the SEAT\_RESERVATION schema.

Referential Integrity Constraint: Flight\_number referring to the Flight\_number in the FLIGHT; Leg\_number referring to the Leg\_number in the FLIGHT\_LEG; and Date referring to the Date in the LEG\_INSTANCE.

Rest attributes have no key nor referential integrity constraint as long as the entity Integrity constraint is not violated.

D.

Flight\_number of FLIGHT\_LEG is FK for Flight.

Departure\_airport\_code and Arrival\_airport\_code of FIGHT\_LEG are FKs for AIRPORT.

Departure\_airport\_code and Arrival\_airport\_code of LEG\_INSTANCE are FKs for AIRPORT.

Airport\_code of CAN\_LAND is FK for AIRPORT.

Flight\_number of LEG\_INSTANCE is FK for FLIGHT.

Leg\_number of LEG\_INSTANCE is FK for FLIGHT\_LEG.

Airplane\_id of LEG\_INSTANCE is FK for AIRPLANE.

Flight\_number of FARE is FK for FILGHT.

Flight\_number of SEAT\_RESERVATION is FK for FLIGHT.

Leg\_number of SEAT\_RESERVATION is FK for FLIGHT\_LEG.

Date of SEAT\_RESERVATION is FK for LEG\_INSTANCE.

Airplane\_type\_name of CAN\_LAND is FK of AIRPLANE\_TYPE.

3.17

Foreign keys:

Serial\_no from OPTION is FK for CAR; parts can be installed to cars with serial numbers.

Serial\_no from SALE is FK for CAR; only car with serial number can be sold.

Salesperson\_id from SALE is FK for SALEPERSON; sales must be performed by a valid salesperson.

Implement:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CAR | Serial\_no | Model | Manufacturer | Price (k USD) |
| 1 | 2017 | Toyota | 18 |
| 2 | 2018 | Toyota | 22 |
| 3 | 2018 | Ford | 24 |
| 4 | 2012 | Ford | 15 |

|  |  |  |  |
| --- | --- | --- | --- |
| OPTION | Serial\_no | Option\_name | Price (USD) |
| 2 | NITRO | 2000 |
| 4 | Turbo charger | 4000 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SALE | Salesperson\_id | Serial\_no | Date | Sale\_price (k USD) |
| 1 | 1 | 2017-09-27 | 15 |
| 2 | 2 | 2017-09-28 | 21 |
| 3 | 3 | 2017-10-01 | 24 |

|  |  |  |  |
| --- | --- | --- | --- |
| SALESPERSON | Salesperson\_id | Name | Phone |
| 1 | John | 9999999999 |
| 2 | Smith | 4444444444 |
| 3 | Snow | 2332332333 |

Example of insertion (w/ violation):

Insert < ‘4’, ‘5’, ‘2016-09-09’, ‘100’ > into SALE is invalid. It violates the referential Integrity constraint. (There is no serial\_no #5 in the CAR table. There is no salesperson with ID = 4, neither.)

Examples of insertion (w/o violation):

Insert < ‘1’, ‘4’, ‘2016-09-09’, ‘100’ > into SALE is valid.

Insert < ‘4’, ‘Jon, ‘6666666666’ > into SALESPERSON is valid.

9.4

Map the EER schema in Figure 8.12 (P278) and Figure 8.7 (P255) into a Relational Data Model. Clearly state the mapping rules and steps.

1. Mapping the regular entity types.

SHIP:



|  |  |  |  |
| --- | --- | --- | --- |
| Sname | Owner | SHIP\_Type | PORT\_Pname |

SHIP\_TYPE:

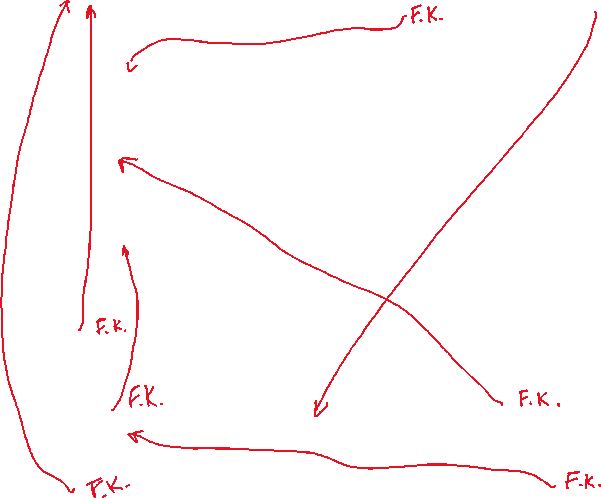
|  |  |  |
| --- | --- | --- |
| Type | Tonnage | Hull |

STATE/COUNTRY:

|  |  |
| --- | --- |
| Name | Continent |

SEA/OCEAN/LAKE:

|  |
| --- |
| SOL\_Name |



2. Mapping the weak entity types.

SHIP\_MOVEMENT:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sname | Date | Time | Longitude | Latitude |

PORT:

|  |  |  |
| --- | --- | --- |
| SOL\_Name | Pname | State\_Name |

PORT\_VISIT:

|  |  |  |  |
| --- | --- | --- | --- |
| Sname | Start\_Date | End\_date | Port\_Pname |

3. Mapping of binary 1:1 relationship types. (in the above tables in THIS color.)

4. Mapping of binary 1:N relationship types. (in the above tables in THIS color.)

5. Mapping of binary M:N relationship types. (in the above tables in THIS color.)

Note: since the relationship is a trinary, step 3 and 5 refers to the same adding of attributes.

6. Mapping of multivalued attributes. (None in this case.)

7. Adding the foreign key referencing.

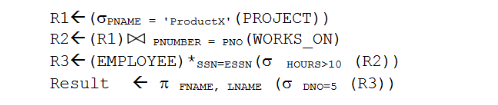
6.2

Union compatibility is a condition that means in any two relations if binary operations (except the cartesian product) are applied on data. It has two constraints: 1. Degree must be the same. 2. Domain of the two tables must be compatible.

Why? If the degrees of the two relations are different, then it is impossible to perform a union operation. If the domain is not compatible, the result table (relation) is meaningless.

6.16

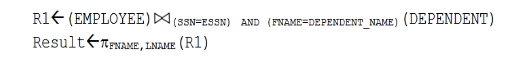
a.



Result:

|  |  |
| --- | --- |
| LNAME | FNAME |
| Smith | John |
| English | Joyce |

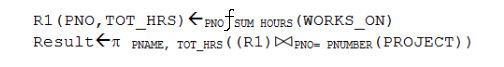
b.



Result:

|  |  |
| --- | --- |
| FNAME | LNAME |
|  |  |

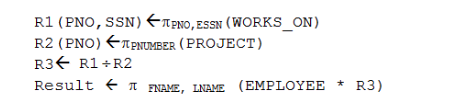
d.



Result:

|  |  |
| --- | --- |
| PNAME | TOT\_HRS |
| ProductX | 52.5 |
| ProductY | 37.5 |
| ProductZ | 50.0 |
| Computerization | 55.0 |
| Reorganization | 25.0 |
| NewBenefits | 55.0 |

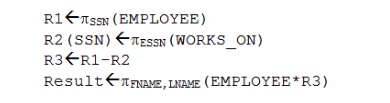
e.



Result:

|  |  |
| --- | --- |
| FNAME | LNAME |
|  |  |

f.



Result:

|  |  |
| --- | --- |
| FNAME | LNAME |
|  |  |

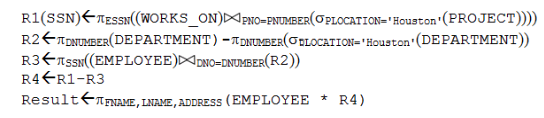
g.

C:\Users\wzwfa\AppData\Local\Microsoft\Windows\INetCache\Content.Word\g.png

Result:

|  |  |
| --- | --- |
| DNUMBER | AVG\_SAL |
| Research | 33250 |
| Administration | 31000 |
| Headquarters | 55000 |

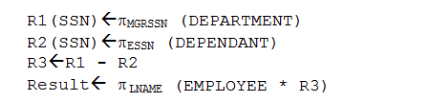
i.



Result:

|  |  |  |
| --- | --- | --- |
| FNAME | LNAME | ADDRESS |
| Jennifer | Wallace | 291 Berry, Bellaire, TX |

j.



Result:

|  |
| --- |
| LNAME |
| Borg |