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.

$$R1 \leftarrow (\sigma_{PNAME = 'ProductX'}(PROJECT))$$
$$R2 \leftarrow (R1) \bowtie_{PNUMBER = PNO} (WORKS\_ON)$$
$$R3 \leftarrow (EMPLOYEE) *_{SSN=ESSN} (\sigma_{HOURS>10} (R2))$$
$$Result \leftarrow \pi_{FNAME, LNAME} (\sigma_{DNO=5} (R3))$$

Result:

LNAME	FNAME
Smith	John
English	Joyce

b.

$$R1 \leftarrow (EMPLOYEE) \bowtie_{(SSN=ESSN) \text{ AND } (FNAME=DEPENDENT\_NAME)} (DEPENDENT)$$
$$Result \leftarrow \pi_{FNAME, LNAME} (R1)$$

Result:

FNAME	LNAME

d.

$$R1(PNO, TOT\_HRS) \leftarrow {}_{PNO}\mathcal{F}_{SUM\ HOURS}(WORKS\_ON)$$
$$Result \leftarrow \pi_{PNAME, TOT\_HRS}((R1) \bowtie_{PNO= PNUMBER} (PROJECT))$$

Result:

PNAME	TOT_HRS

ProductX	52.5
ProductY	37.5
ProductZ	50.0
Computerization	55.0
Reorganization	25.0
NewBenefits	55.0

e.

$$R1(PNO, SSN) \leftarrow \pi_{PNO, ESSN}(WORKS\_ON)$$
$$R2(PNO) \leftarrow \pi_{PNUMBER}(PROJECT)$$
$$R3 \leftarrow R1 \div R2$$
$$Result \leftarrow \pi_{FNAME, LNAME}(EMPLOYEE * R3)$$

Result:

FNAME	LNAME

f.

$$R1 \leftarrow \pi_{SSN}(EMPLOYEE)$$
$$R2(SSN) \leftarrow \pi_{ESSN}(WORKS\_ON)$$
$$R3 \leftarrow R1 - R2$$
$$Result \leftarrow \pi_{FNAME, LNAME}(EMPLOYEE * R3)$$

Result:

FNAME	LNAME

g.

$$R1(DNUMBER, AVG\_SAL) \leftarrow {}_{DNO}\mathcal{F}_{AVG\ SALARY}(EMPLOYEE)$$
$$Result \leftarrow \pi_{DNUMBER,\ AVG\_SAL}(R1\ *\ DEPARTMENT)$$

Result:

DNUMBER	AVG_SAL
Research	33250
Administration	31000
Headquarters	55000

i.

$R1(SSN) \leftarrow \pi_{ESSN}((WORKS\_ON) \bowtie_{PNO=PNUMBER}(\sigma_{PLOCATION='Houston'}(PROJECT))))$

$R2 \leftarrow \pi_{DNUMBER}(DEPARTMENT) - \pi_{DNUMBER}(\sigma_{DLOCATION='Houston'}(DEPARTMENT))$

$R3 \leftarrow \pi_{SSN}((EMPLOYEE) \bowtie_{DNO=DNUMBER}(R2))$

$R4 \leftarrow R1 - R3$

$Result \leftarrow \pi_{FNAME, LNAME, ADDRESS}(EMPLOYEE * R4)$

Result:

FNAME	LNAME	ADDRESS
Jennifer	Wallace	291 Berry, Bellaire, TX

j.

$R1(SSN) \leftarrow \pi_{MGRSSN}(DEPARTMENT)$

$R2(SSN) \leftarrow \pi_{ESSN}(DEPENDANT)$

$R3 \leftarrow R1 - R2$

$Result \leftarrow \pi_{LNAME}(EMPLOYEE * R3)$

Result:

LNAME
Borg