Abhinav Yalamaddi - YXA210040

Database Design - CS 6360.002 - Fall 2022 - HW2A

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

Domain - A domain D is a set of atomic values, which means that each value in the domain is indivisible.

Attribute - An attribute is a feature or property of an entity. A relation schema consists of a relation name and a list of attributes. Each attribute represents the name of domain D's role in the relation model.

N-tuple - It is an ordered list of values, where each value is either a domain member of A or a special NULL value.

Relation schema - A relational schema represents a table's name as well as its characteristics. It describes a relationship.

Relation State - A relation state is a subset of the cartesian product of the domains that define the relation schema.

Degree of relation - It is the nth attribute in a relation schema. It is also known as the arity of the relationship. Nothing but cardinality.

Relational database schema - A collection of definitions in the table that are used in the derived views. It is a template for how data is structured.

Relational database state - The set of integrity constraints and the relation schema set are a collection of relation states that meet the integrity constraints.

3.3. Why are duplicate tuples not allowed in a relation?

Duplicate tuples are not permitted in a relationship because they produce data redundancy and break the restrictions of relational integrity. They slow down database updates and impair database processing speed.

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

A key is a single or multiple-field combination. The goal is to access or get data rows from a table as necessary. Tables define keys to rapidly and easily retrieve or sequence the stored data. They can also be used to connect distinct tables.

A superkey is a collection of attributes that may be used to uniquely identify a database record. A table may include a large number of superkeys. Candidate keys are a subset of superkeys that do not include any superfluous information.

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

Entity integrity constraint: No primary key value may be null. Individual tuples in a relation are identified using this. We cannot identify some tuples if the main key has null values.

Referential integrity constraints are defined between two relations and are used to preserve consistency between tuples in the two relations. It guarantees that the connection between the transactions table and the parts table is formed correctly. The item number forms the foreign key in the transaction table, which is an attribute that ties to the primary key in the parts table. Only item numbers from the components table are permitted to appear in the transaction table. This is referred to as referential integrity.
Entity integrity is referred to as the main key, whereas referential integrity is referred to as the foreign key.

3.9. Define foreign key. What is this concept used for?
A foreign key is a column or set of columns in a relational database table that connects data from two tables. Because it refers to the primary key of another table, it works as a cross-reference between both.
A foreign key must always be associated with a primary key. The referred table is the original table that contains the main key. Multiple foreign keys from other tables can refer to this key.

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.

b. Insert <‘ProductA’, 4, ‘Bellaire’, 2> into PROJECT.
This violates the DNUM's referential integrity requirement since no DEPARTMENT tuple exists with Dnumber 2. We may enforce this requirement by inserting the tuple with DNUM=2 or inserting the tuple in DEPARTMENT with Dnumber=2.

c. Insert <‘Production’, 4, ‘943775543’, ‘2007-10-01’> into DEPARTMENT.
This violates the key constraint since another tuple 'Administration' with the same Dnumber already exists. To enforce the restriction, the tuple might be assigned to a DEPARTMENT with a different Dnumber.

d. Insert <‘677678989’, NULL, ‘40.0’> into WORKS_ON.
Because the main key is NULL, this violates the entity integrity restriction. It is possible to ensure this by assigning a unique value to the main key.

e. Insert<‘453453453’, ‘John’, ‘M’, ‘1990-12-12’,‘ spouse’>intoDEPENDENT.
This tuple meets all of the requirements. As a result, it is accepted in the table.

g. Delete the EMPLOYEE tuple with Ssn = ‘987654321’.
This is not acceptable since the tuples WORKS_ON, DEPENDENT, and DEPARTMENT all relate to this tuple. That is, they have this ssn value as a foreign key.
To guarantee that the tuple may be destroyed, we must first delete all the tuples in other tables that relate to it.

k. Modify the Hours attribute of the WORKS_ON tuple with Essn = '999887777' and Pno = 10 to '5.0'.
This meets all of the criteria and is therefore acceptable.

3.12. Consider the AIRLINE relational database schema is 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.

a. Give the operations for this update.
Assuming that seats are available on the desired aircraft or flight segment, the update action may be conducted by putting a new tuple into the SEAT RESERVATION table.
Updated Query:
Insert into SEAT_RESERVATION values <'GTA11', 1, '2022-10-09', '99A', 'Abhinav', '9876543210'>

b. What types of constraints would you expect to check?
If the requested flight or flight leg has non-reserved seats available on the specified date. If the specified flight or flight leg is available on the specified date.

c. Which of these constraints are key, entity integrity, and referential integrity constraints, and which are not?
Referential integrity: If the desired flight or flight leg is available on the specified date, referential integrity is maintained. The flight number in the FLIGHT database may be used to verify this.
Entity integrity: A non-reserved seat must be available on the specified date to verify that the reservation has a unique key in the SEAT RESERVATION database.

d. Specify all the referential integrity constraints that hold on the schema shown in Figure 3.8.
An FK value in a tuple t1 is either a PK in some tuple t2 in the relation or has the value NULL. In the case of t1[FK]=t2[PK], tuple t1 is said to refer to tuple t2.
We can see that the foreign keys in all of the tables correspond to the primary key values in some of the relation's tuples. FLIGHT LEG, FARE, and LEG INSTANCE, for example, are the FKs pertaining to FLIGHT.

Abhinav Yalamaddi
YXA210040

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.

Foreign Keys:

1. Serial no in OPTION is the FK referring to Serial no in CAR.
2. Serial no is the FK referring to CAR in SALE.
3. The FK referencing to SALESPERSON in SALE is salesperson id.

INSERT into CAR values<1234, ‘CRV’, Honda, 25000>
INSERT into OPTION values<1234, ‘Heater’, 895>
INSERT into SALESPERSON values<9, ‘Jack’, ‘7788996677’>
INSERT into SALE values<1, 1234, ‘2022-10-09’, 30000>

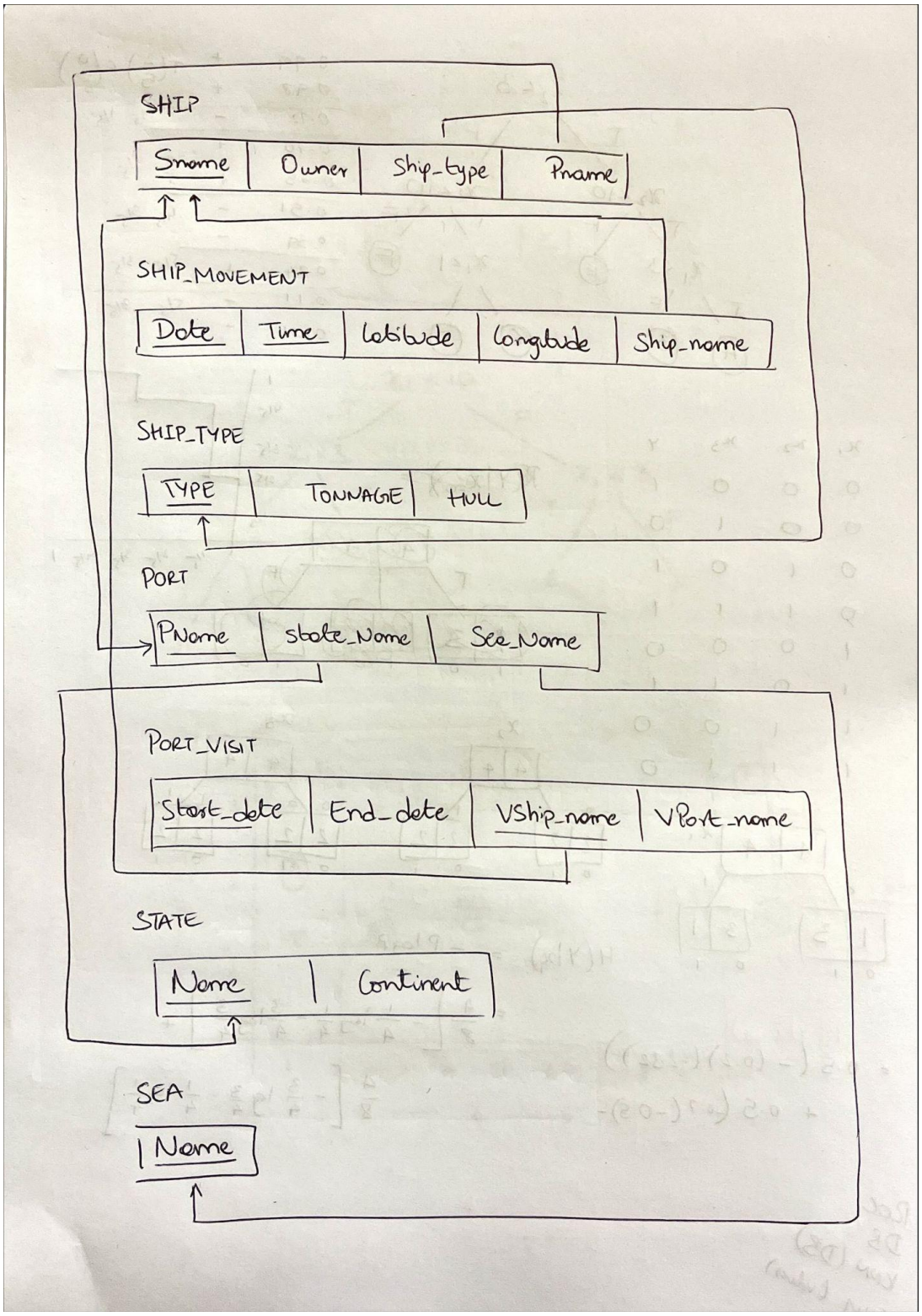
Referential integrity restrictions are violated: for example, Inserting into the SALES database, INSERT into SALE values<1, 1234, ‘2022-10-09’, 50000>

Because the Salesperson id added in SALE does not exist in the SALESPERSON database, there is no reference to the PK, this violates the referential integrity restrictions.

Don’t violate:
INSERT into SALE values<1, 1234, ‘2022-10-09’, 28945>
The referential integrity criterion is not violated since the Salesperson id refers to the tuple SALESPERSON, which has an identical value. Serial no refers to a valid no in the CAR database.

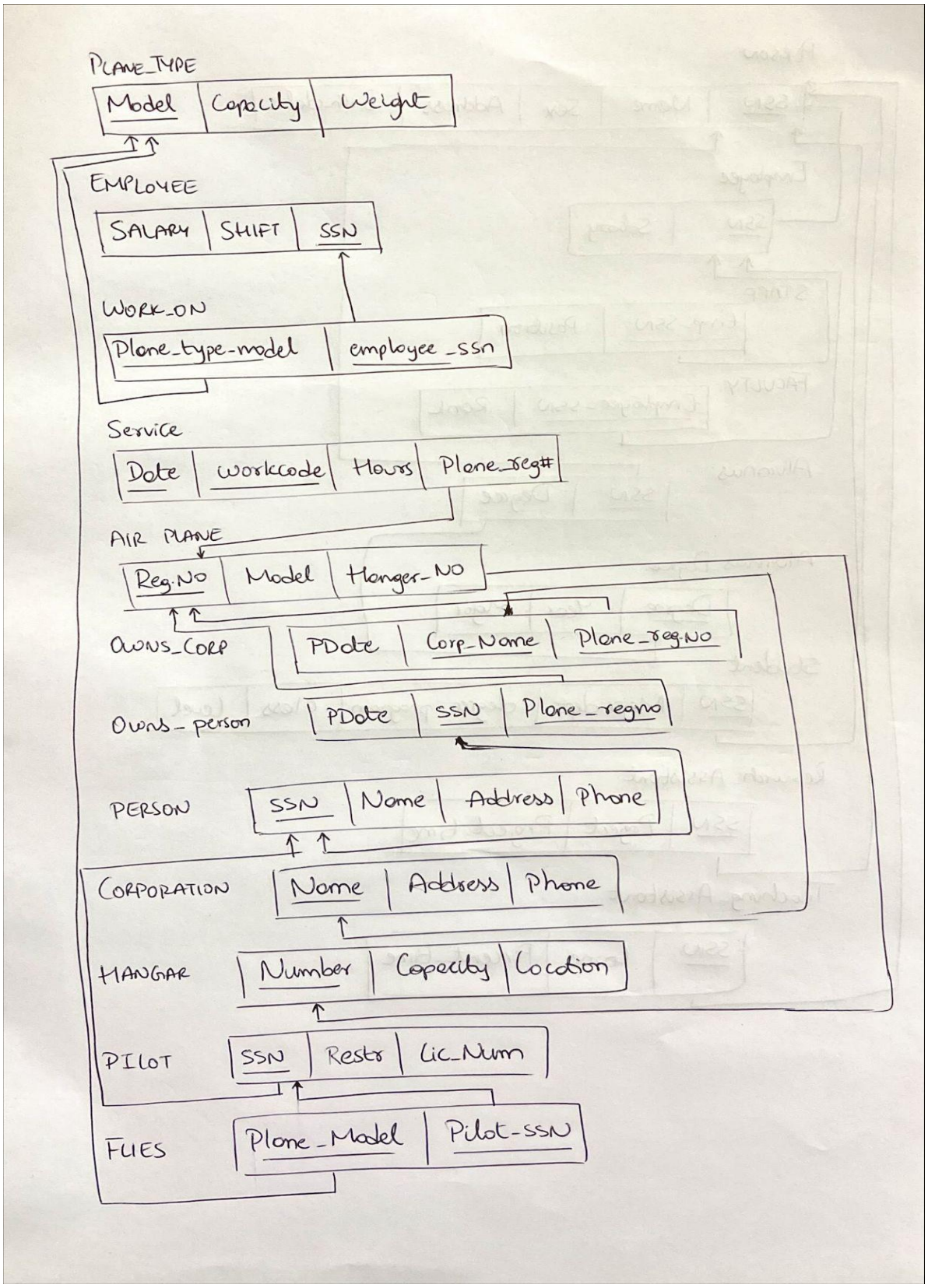
Abhinav Yalamaddi
YXA210040

9.4. 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.



Abhinav Yalamaddi
YXA210040

Extra 1: Map the EER schema in Figure 8.12 (P278) into a Relational Data Model



Abhinav Yalamaddi
YXA210040

Extra 2: Map the EER schema in Figure 8.7 (P255) into a Relational Data Model