Assignment - 04

Girish Khule
23PGAI0057

26/9/2022

Ques. 3.

Sol^n - F.D. in the ~~dependencies~~ are

RegistrationNumber → modelNumber
modelNumber → capacity.
aadharNumber → address, name, phoneNumber, salary
testNumber → name, maximumScore
testNumber, registrationnumber, aadharnumber →
date, noofhours, busScore.

Suppose: registrationNumber, modelNumber, aadharnumber, testNumber are the primary key.

The given description contains various ~~department~~ dependencies and for creating the relational schema we have to remove them from the database.

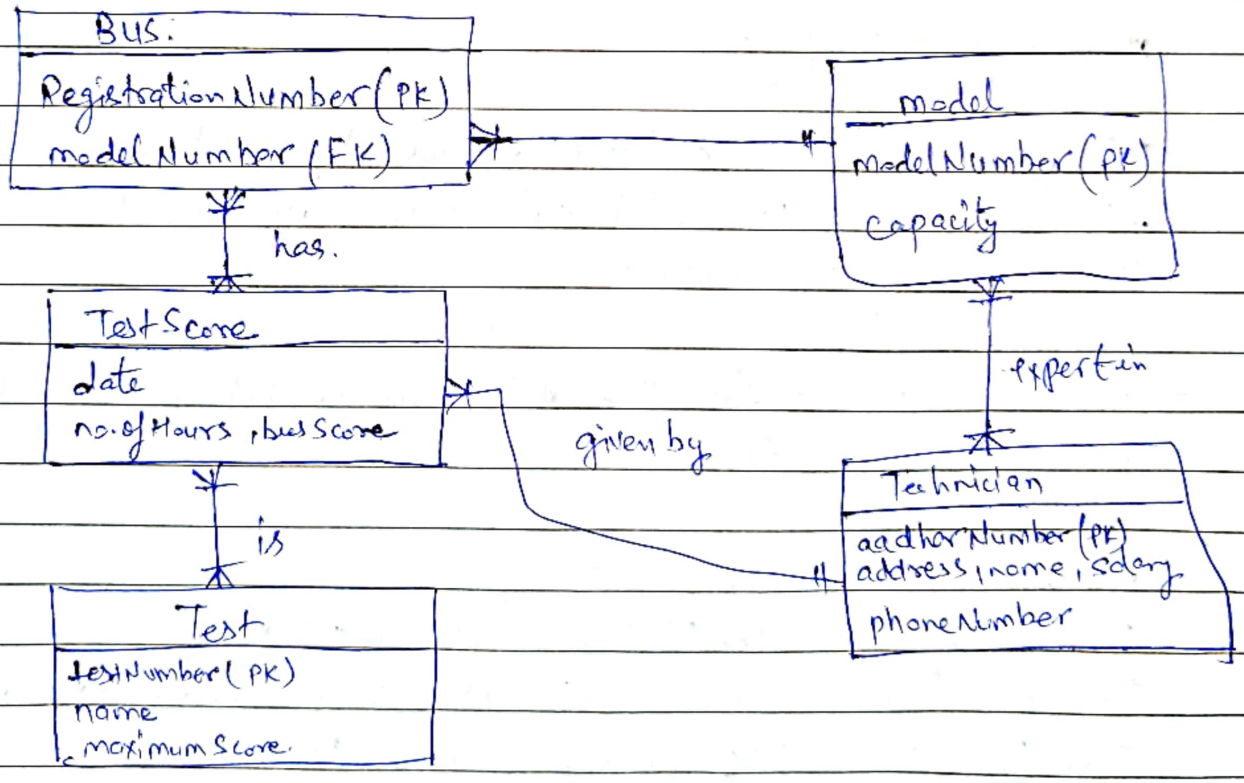
Relational schema for the description in 3rd normalized form contain PK's and FK's.

mode (modelNumber(PK), capacity)
Bus (RegistrationNumber(PK), modelNumber(FK))
Technician (aadharNumber(PK), address, name, phoneNumber, salary
Test (testNumber(PK), name, maximumScore)
TestScore (date, noofHours, busScore, testNumber(FK), RegistrationNumber(PK), aadharNumber(FK))
Expert (aadharNumber(FK), modelNumber(FK)).

* Relationship b/w tables are.

1. There are 1:M relationship between model tables and bus table
2. There are M:1 relationship between table and Bus table.
3. There are M:N relationship between Technician table and model table.
4. There are M:1 relationship between testScore table and test Table.
5. There are 1:M relationship between technician table and test score table.

* Entity relation diagram for the given relational schema.



__/__/____

Ques. 1 - Explain the following term briefly along with an example. Trivial functional dependency, Non-trivial functional dependency Transitive functional dependency, 1NF, 2NF, 3NF and BCNF.

Sol^{n} - Trivial functional dependency

It is a functional dependency where -

- If B is a subset of A then A → B has a trivial functional dependency
- The following dependencies are also trivial like A → A, B → B.

Example -

A table containing the columns Students_id and Student_Name is an example.

{Student_id, Student_Name} → Student-id is a trivial functional dependency, as Student_id is a subset of {Student-id, Student_Name} That makes sense because if we know the value of Student-id and Student_Name then the value of Student-id can be uniquely determined. Also, Student_id → Student-id and Student-Name → Student_Name are trivial dependency too.

- Non-trivial functional dependency

It is functional dependency where -

- If B is not a subset of A then A → B has trivial dependency
- When A intersection B is NULL then A → B is called as complete non trivial.

Example

D → Name

Name → DoB

- Transitive functional dependency -

Transitive dependency is the term for any indirect link that results in functional dependency (FC), if

A → B and B → C are true then A → C happens to be a transitive dependency.

Thus, the transitive dependency must be removed in order to obtain 3NF.

Example –

Book → Author-Nationality. If we known the book name, we can determine the authors nationality via the author col.

* 1NF

- If relation has atomic value, it is 1NF, it states that a table's attribute cannot have more than one value, it can only contain characteristics with single values. Multi-valued characteristics, composite attributes and their combinations are not allowed in the first normal form.

Requirements →

The following are explanation of the requirements to be taken into account when designing 1NF.

- A primary key exists for each table (a minimal set of attributes which can uniquely identify a record)
- A table's column contain atomic value (No multi-valued attributes allowed)
- There are no grouping that repeat (two column do not store similar information in the same table)

Example.

Relation EMPLOYEE is not in 1NF because of the multivalued attributes EMP_PHONE

*. 2-NF.

When a relation is in first normal form but doesn't have any non-prime attributes that are functionally dependent on any candidate key suitable subset, it is said to be in a 2NF in a relational database.

- In the 2NF, relational must be in 1NF
- In the second normal form, all non-key attributes are fully functionally dependent on the primary key

Example

Let's say that a school has a database where instructor's information and the subjects they teach are stored, A teacher in school is allowed to teach multiple subjects.

* 3-NF.

When a given relation is in 2NF but lacks as transitive, partial dependency, it is said to be in its 3NF

- If relation is in 2NF and doesn't have any transitive partial dependency it will be 3NF
- The amount of duplicate data is decreased with 3NF Additionally, it is employed to ensure data integrity
- The relation must be in third normal form if non-prime characteristics do not have transitive dependencies.

Example. A Relation is in 3NF if it holds at least one of the following conditions for every non-trivial function dependency $X \rightarrow Y$.

1. X is super key
2. Y is prime attribute i.e each element of Y is part of some candidate key

__/__/____

* BCNF

The Boyce Codd Normal form, commonly referred to as the 3.5 Normal form, is an improved variant of the third normal form.

- The upgraded version of 3NF is BCNF. Compared to 3NF, it is harsher harsher
- If X is the super key of the table and every functional dependency XY, the table is said to be in BCNF
- The table should be in 3NF for BCNF and LHS is super important for every FD.

Example.

F: {Student, Teacher} → subject
{Student, subject} → teacher
Teacher → subject

Ques) – Given the relation R = {P, Q, R, S, T, U, V, W, X, Y, Z} and the set of functional dependencies F = {(P,R) → {QR}, {PQ} {P} → (S,T), {R} → {U}, {U} → {V,W}, (S) → (X,Y), (U) → (Z). Find the key for (R)? Decompose (R) into 2NF and then 3NF relation and then to BCNF.

Ans → The key is {P,R) because it is the only candidate key and it is the only one that satisfies the requirement for a being a key.

R1 is not in 3NF because there is a functional dependency (P,R) → (Q) where (Q) is not a subset of the key

We decompose R1 into relation R11 = {P, Q, R}
R12 = {P, S, T}

Sol^{n}. – The key is (P, R).

We start with the relation R = {P, Q, R, S, T, U, V, W, X, Y, Z}

R is not in 2NF because there is a functional dependency (P) → (S, T) where (S, T) is not a subset of the key.

We decompose R into relations
R1 = (P, Q, R, S, T)
R2 = (P, U, V, W, Z) which are both in 2NF.

R1 is not in 3NF because there is a functional dependency (P, R) → (Q) where (Q) is not a subset of the key.

We decompose R1 into the relation R11 = (P, Q, R)
R12 = (P, S, T) ~~in 3~~

R2 in 3NF.

R11 and R12 are in BCNF because they are in 3NF and every functional dependency is a dependency on the key.

Therefore the key for R is (P, R)