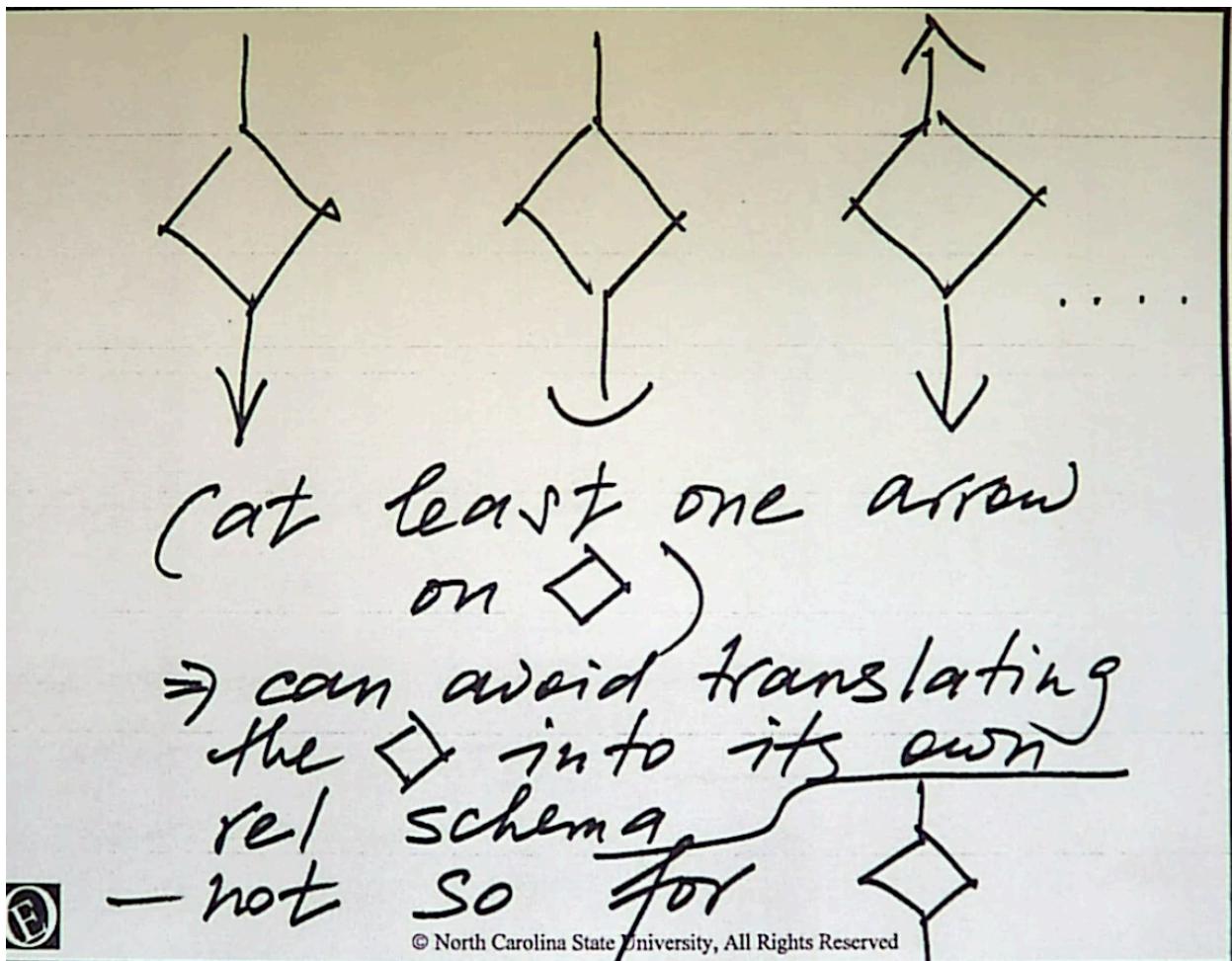
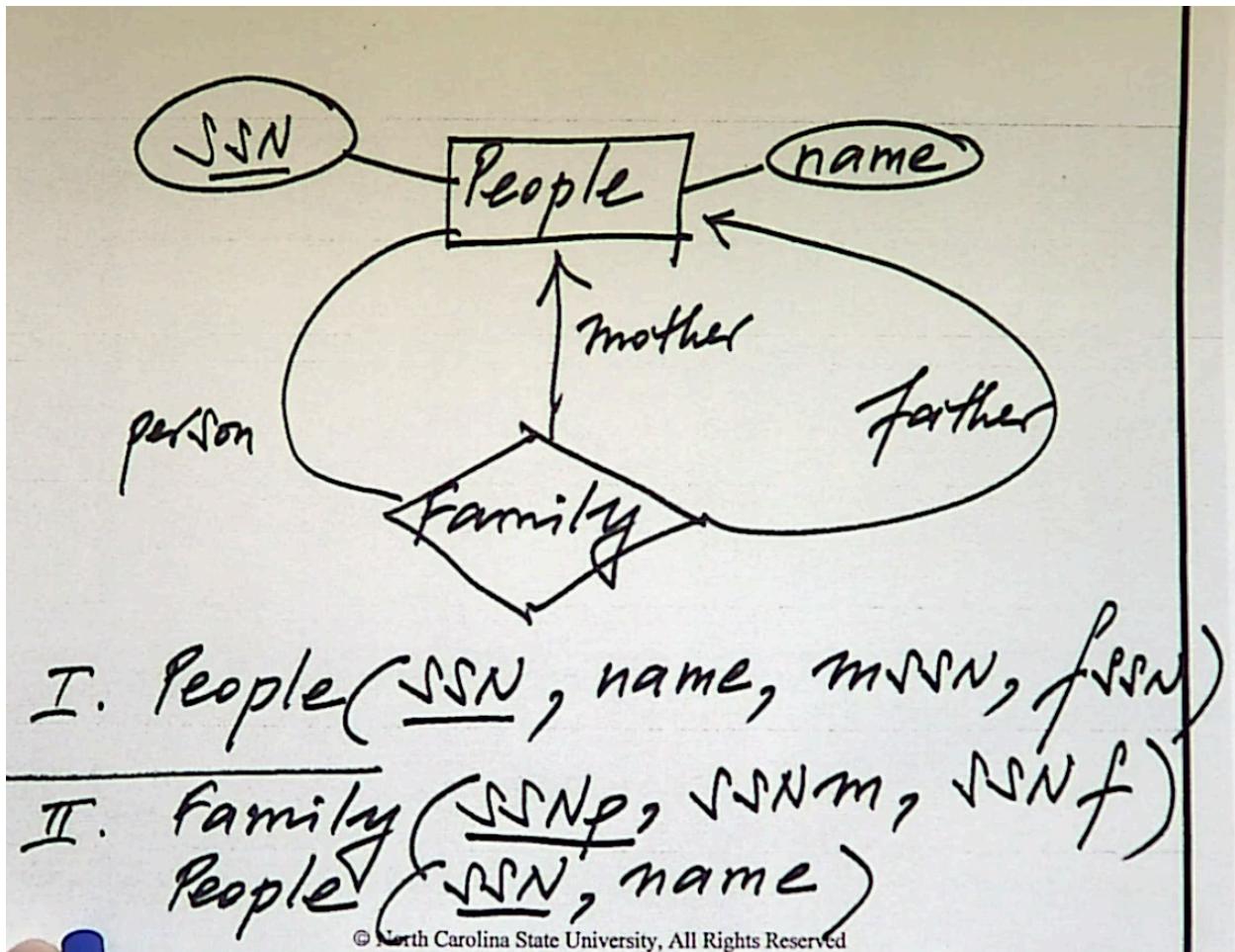


08/31/2023

Translation of E/R into Relations



In a Many-One relation, make a relation only for the many side and fit all attributes in the Many side. Skip the relation diamond in between.



While translating roles, for the diamond, write the key of the diagram for every possible role.

Weak entities- key for itself and the key for supporting entity

Subclasses should only be defined when there is at least one extra attribute or relation than the root entity

Strategies for translating Hierarchies:

Nulls approach - single relation containing all attributes

E/R approach

Object Oriented approach - Even if two classes have same relation schema, doesn't matter. It infact shows they are disjoint and need no Join operations

09/05/2023

FDs are defined on schemas and not on data

In a FD, if A->B, then for a given value of A, there can be a single value of B

When there are bad relational schemas

Update anomaly - If there are multiple rows with redundant (same) attributes, and lets say one attribute needs to be updated. Then obviously all rows where this attribute existed, need to be updated.

Deletion anomaly- Lets say High school code -> High school city. We have information about an applicant and his high school info. Now if no other applicant studied at the same school and if this record needs to be deleted, we will lose the information about the high school code and high school city.

Making ER diagrams first and then generating relation schemas from it solves this issue=> Good design

Superkeys - attribute that defines all other attributes. Can be reduced to remove redundant attributes

Keys - Minimal superkey. The one where no more attributes can be removed, otherwise not all other attributes will be defined.

Does any of the subset of a Superkey give all attributes? If yes, then not a key, else a Key

Closure of (A) = A + All other attributes defined by A

09/07/2023

E/R keys don't have requirement for minimality like for relational keys

E/R keys	Relational Keys
Properties of entities	Properties of tuples
One tuple corresponds to one entity usually so its the same	In poor relational schemas, one entity can become several tuples. So they are different

E/R keys give FDs from entity set keys and many one relationships

A FD is trivial if it does not have any new attributes on the right hand side

Derived FDs: All new FDs that you discovered except the trivial FDs

When you are writing Derived FDs for Projection based questions, just remove the ones which are trivial and having irrelevant attributes. Write others even if they are repeated in the question.

BCNF-> When all FDs of a relation schema have a superkey on the LHS
If this is violated by any FD, you need to split the schema.

How to split?

Take a FD which is in violation.

R1 - Closure of LHS of FD

R2 - LHS of FD + all remaining attributes

BCNF-> No redundancy, No update and delete anomalies, Lossless

09/12/2023

How to make joins on relations using values given

A 2 attribute relation is always in BCNF

If we have no FDs we have no problem, the schema is always in BCNF

BCNF vs 3NF what is preserved and what is not

3NF is satisfied when it is in BCNF or

RHS of FD is a part of the key

3NF gives you Recovery and Dependency preservation and not necessarily elimination of Redundancy, update and deletion anomaly

BCNF gives you recovery, elimination of redundancy, update and deletion anomalies and not necessarily dependency preservation

09/14/2023

Schema of a relation -> list of all its attributes

Projection - Selecting certain attributes

Condition

Cross Product - Left inputs glued to right inputs

Theta Join - Cross Product followed by Selection

Natural join - Concatenating on the same values between two relations. Implied condition - Same name attributes have same values. Drop duplicates on same name attributes

Removing duplicates from Projection is very expensive ($n \log n$), so SQL does not do this. It keeps the duplicates in something called as bags

Bags- Duplicate elements are allowed

Renaming- Same as sets

Selection- same as sets

Projection- we dont remove duplicates as done for sets

Product- same as sets

Theta Join- same as sets

Natural Join- we dont remove duplicates as done for sets

Set operators- different: duplicates can arise

Operations in relational algebra (core) on sets of tuples

- * Relation by name: R
- * Set operators:
 $R \cup S, R \cap S, R - S$
- * Renaming: $\rho_{T(K, L)}(S)$
- * Selection: $\sigma_C(R)$
Boolean expression:
can use
AND, OR,
NOT

© North Carolina State University, All Rights Reserved

Schema-Defining Rules 1



- ◆ For union, intersection, and difference, the schemas of the two operands must be the same, so use that schema for the result.
- ◆ Selection: schema of the result is the same as the schema of the operand.
- ◆ Projection: list of attributes tells us the schema.

Schema-Defining Rules 2

- ◆ Product: the schema is the attributes of both relations.
 - ◆ Use R.A, etc., to distinguish two attributes named A.
- ◆ Theta-join: same as product.
- ◆ Natural join: use attributes of both relations.
 - ◆ Shared attribute names are merged.
- ◆ Renaming: the operator tells the schema.

28

Relational algebra - procedural
SQL - non procedural

While using aggregator functions, only non null values are considered

δ - eliminating duplicates

τ - sorting - potentially on several attributes
⇒ the output is a list

π generalized - arithmetic, duplicate columns, etc

ρ - grouping and aggregation.
= standard fns: COUNT,

SUM, MIN, MAX, AVG,

© North Carolina State University, All Rights Reserved

[CNTD]

~~\bowtie~~ - outer join

$R \bowtie S$: natural join

$R \bowtie^o S$: natural full outerjoin

$\underline{R} \bowtie_L S$: natural left outerjoin

$R \bowtie_R \underline{S}$: natural right outerjoin

$$R \dot{\times} S = R \times S$$

$R \bowtie_{R.B < S.B} S$ full theta outerjoin

09/26/2023

so far

- > subqueries
- ~~the~~ set operators
- IN, NOT IN
- EXISTS, NOT EXISTS
- ALL, ANY

NULL's Ignored in Aggregation

- ◆ NULL never contributes to a sum, average, or count, and can never be the minimum or maximum of a column.
- ◆ But if there are no non-NULL values in a column, then the result of the aggregation is NULL.

Illegal Query Example

- ◆ You might think you could find the bar that sells Bud the cheapest by:

```
SELECT bar, MIN(price)  
FROM Sells  
WHERE beer = 'Bud';
```

- ◆ But this query is illegal in SQL.
 - ◆ Why? Note bar is neither aggregated nor on the GROUP BY list.

*SELECT A, B , SUM(F)
FROM R
GROUP BY A, B, C, D*

If you mention A in Select clause, it has to be in Group BY clause

The condition of Having clause applies to each group given by GROUPBY

Post Midterm

10/17/2024 + 10/19/2023 - Constraints and Triggers

CONSTRAINTS

Constraints on feeded values in SQL

Why use Constraints

Data entry errors

DB modifications: correctness criteria

To enforce consistency across DB data

Kinds of constraints:

Primary Keys

2 rows cannot have same values of Primary key attribute

UNIQUE

UNIQUE mei Null chalta hai

UNIQUE(name, addr) and UNIQUE(name), UNIQUE(addr)

Are different

NOT NULL

Foreign keys

Foreign key values can have NULL

Referenced (Jisko reference karte hai vo)- Unique key/Primary key hona chahiye

Disallow updates or inserts in the Referencing relation

Deletions are not a problem in Referencing relation

Inserting in referenced not a problem

Deletions and Updates in the referenced relation are a problem if some pointers are pointing to them. By default, these are not allowed in the system

ON UPDATE CASCADE If referenced relation is updated, referencing relation is also updated

ON UPDATE SET NULL

ON DELETE CASCADE If referenced relation is deleted, referencing relation is also deleted

ON DELETE SET NULL

Attribute based

CHECK should be used for attribute based checks

Attribute based checks are performed only when there are Insert/ updates.

```
CREATE TABLE xyz
(
Age INT CHECK (Age < 100)
);
```

Tuple based

Tuple based constraints - Relations between values in a single row

Are performed only when there are Insert/ updates.

```
CREATE TABLE xyz
(a,
b,
c,
CHECK (a+b<=24)
);
```

A->B = ~A OR B

Tuple based checks can be used to express the full power of Foreign key constraints and Functional Dependencies.

Assertions

First class citizens like Create table and Create Trigger statements -> You can Alter, Create, Delete assertions like you can do with Tables.

They only check if the data is consistent but do not modify the data

Executed when data is updated, inserted or deleted

Not used in most of the DBMS because they are expensive in implementation

CREATE Assertion <name> CHECK (<condition>);

TRIGGERS

Similar to Assertions but they also have the power to modify the data.

Has 3 components: EVENT, CONDITION, ACTION

```
CREATE TRIGGER <name>
AFTER .... => event
REFERENCING NEW AS <>
WHEN <condition>....=> condition
UPDATE .... => action
FOR EACH ROW;
```

You can also use

- =>REPLACE TRIGGER
- =>BEFORE (instead of AFTER)
- =>INSERTS, UPDATES,DELETES (in action)
- =>OLD (instead of NEW when you are maybe trying to delete something)
- =>More than 1 SQL statements in the action part using BEGIN..END

If campus enrollment increases from below 7000, delete all "new" applications to the campus (dated after 2/15/20), and set all 'Y' decisions for applications before 2/15/20 to 'U'

CREATE OR REPLACE TRIGGER TooMany
event [AFTER UPDATE OF enrollment]
ON campus
REFERENCING OLD AS OldVal
cond [WHEN (OldVal.enrollment < 7000
AND NewVal.enrollment >= 7000)]
[NEW AS NewVal]
[© North Carolina State University, All rights reserved.]

BEGIN

DELETE FROM Apply
WHERE location = New Val. location
AND date > 2/15/20;

UPDATE Apply

SET decision = 'U'
WHERE location =
old Val. location
AND decision = 'Y'

END

FOR EACH Row;

Options: FOR EACH ROW

- ◆ Triggers are either *row-level* or *statement-level*.
- ◆ FOR EACH ROW indicates row-level; its absence indicates statement-level.
- ◆ Row level triggers are executed once for each modified tuple.
- ◆ Statement-level triggers execute once for an SQL statement, regardless of how many tuples are modified.

31

10/24/2023 => in notebook

10/26/2023

Transaction T:

- looks at PC data
 - = speeds
 - = prices
 - runs "forever"
 - at each hour checks whether there is a PC that
 - = has speed ≥ 1500
 - = and sells for $< \$1K$
- if T finds such a PC - then update its info

© North Carolina State University, All Rights Reserved

Think about all 4 Isolation levels for the above example. Discuss in office hours

Other notes in Notebook