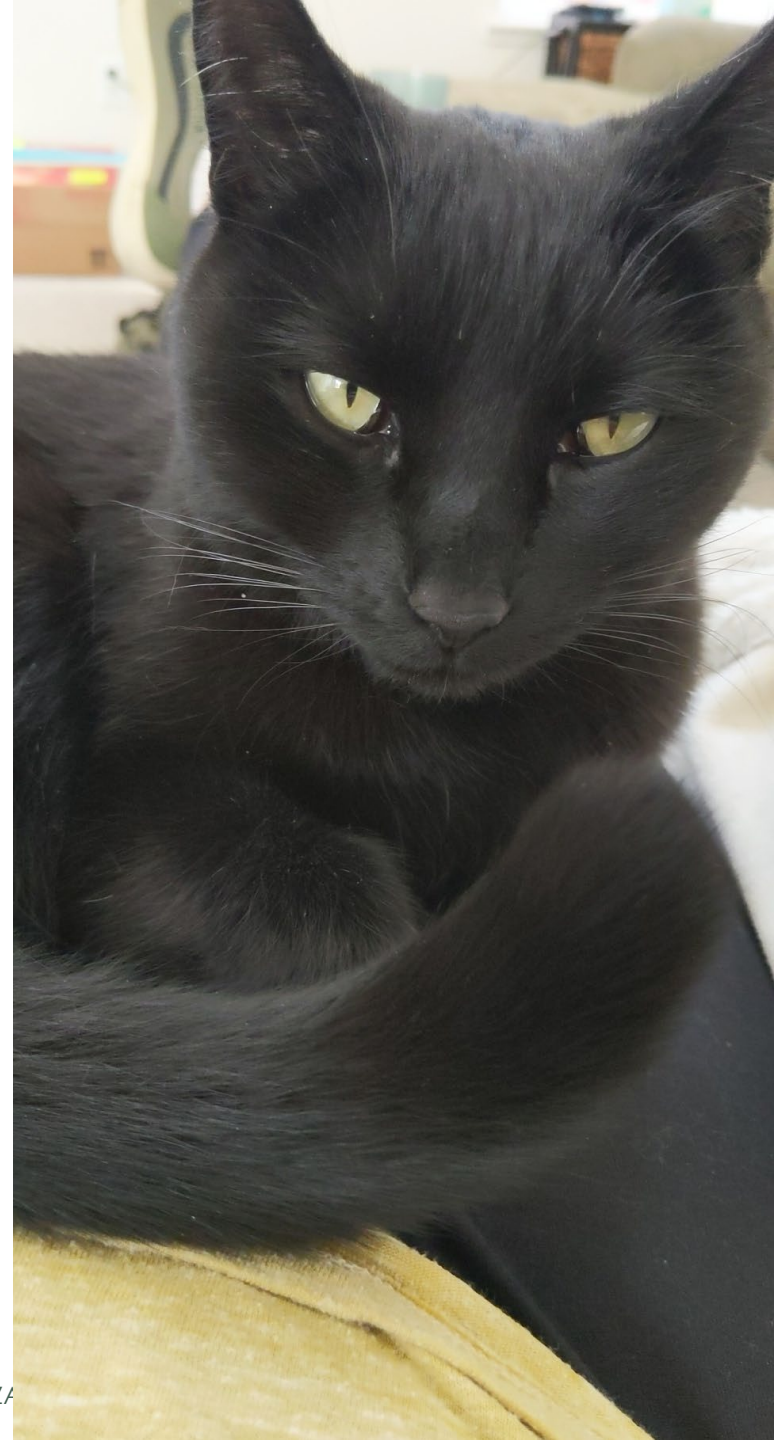

Design

ERIN KEITH

Goals

1. Referential Dependencies and Keys
2. Normalization
3. Practice!

Relations

- order of tuples doesn't matter
- relations (tables) change
  - insert new tuple
  - delete existing tuple
  - update existing tuple
- schemas are ***not*** expected to change!

Keys

a set of attributes forms a **primary key** for a relation

Movies(<u>title</u>, <u>year</u>, length, genre)

- no two tuples in a relation instance to have the same values in all the attributes of the key

Keys

- cannot be **`NULL`**
- the set of keys must be *unique*
  - otherwise the insert operation will result in an error

```
CREATE TABLE MovieStar (
    name CHAR(30) PRIMARY KEY,
    address VARCHAR(255),
    gender CHAR(1),
    birthdate DATE
);
```

OR

```
CREATE TABLE MovieStar (
    name CHAR(30),
    address VARCHAR(255),
    gender CHAR(1),
    birthdate DATE,
    PRIMARY KEY (name)
);
```

Keys

- cannot be **NULL**
- the set of keys must be *unique*
  - otherwise the insert operation will result in an error

```
CREATE TABLE Movies (
    title       CHAR(100),
    year        INT,
    length      INT,
    genre       CHAR(10),
    studioName  CHAR(30),
    producerC#  INT,
    PRIMARY KEY (title, year)
);
```

Constraints

- **Referential-Integrity Constraint**
  - a value appearing in some attribute or set of attributes must also appear in the corresponding attribute(s) of some tuple of the same or another relation
  - use a **`REFERENCES`** or **`FOREIGN KEY`** declaration in the relation schema

Foreign Key

- cannot be **NULL**
- indicates a dependency on another relation's primary key
- a value appearing in one context also appears in another, related context

Example 2.21: Consider the two relations from our running movie database:

```
Movies(title, year, length, genre, studioName, producerC#)
MovieExec(name, address, cert#, netWorth)
```

Constraints

- **Attribute-Based Check Constraint**
  - on the value of an attribute
    - `presC# INT REFERENCES MovieExec(cert#) NOT NULL`
    - avoids the system trying to fix foreign-key violations by making the missing attribute be **`NULL`**
  - add the keyword **`CHECK`** and the condition to be checked after the declaration of that attribute in its relation schema
    -
```
presC# INT REFERENCES MovieExec(cert#)
          CHECK (presC# >= 100000)
```
    - requires that certificate numbers be at least six digits

Dependencies

Functional Dependency

- A functional dependency is a relationship between two sets of attributes in a database, where one set (the determinant) determines the values of the other set (the dependent).
  - Example: primary key

Dependencies

Full Functional Dependency

- dependent attributes are determined by the determinant attributes
- Example:

  employee ID determines employee Name and Address

Dependencies

Partial Functional Dependency

- dependent attributes are partially determined by the determinant attributes
- Example:

  employee ID determines employee Name but not necessarily Address

Dependencies

Transitive Functional Dependency

- dependent attributes are determined by a set of attributes that are not included in the determinant attributes
- Example:

  employee ID determines employee Department which determines employee Salary

Normalization

First Normal Form

- 1NF
- every table has a primary key
- all data is atomic

- Atomic Example:

address with city and state or address, city, state

Normalization

Second Normal Form

- 2NF
- table is in 1NF
- all non-primary key attributes are functionally dependent on the primary key

- Example:

Employee ID -> Employee Name, Address, etc.

Normalization

Third Normal Form

- 3NF
- the table is in 2NF
- all non-primary key attributes are not functionally dependent on any non-primary key attributes

- Example:

Tournament winners

Tournament	Year	Winner	Winner's date of birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

Normalization

Third Normal Form

- 3NF
- the table is in 2NF
- all non-primary key attributes are not functionally dependent on any non-primary key attributes

- Example:

Tournament winners

Tournament	Year	Winner
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson

Winner's dates of birth

Winner	Date of birth
Chip Masterson	14 March 1977
Al Fredrickson	21 July 1975
Bob Albertson	28 September 1968

Next Class

Module:

Week 4: Background, Ch 4

Topic:

ERDs to UML