Database Revision

Problems with file based systems

Data redundancy

Data isolation

Integrity problems

Security problems

Concurrency access

Data modelling has the aim to describe:

The data conained in the database (entities: students, lecturers, courses)

The relationship between data items (students supervised by lecturers; lecturers teach courses)

The constraints of data (student number has exactly 8 digits, a subject has four of 6 units of credits only)

Relational data model

Relation – table

Row/record – represents a group of related data values

Column/attribute – is a field.

Degree – is the number of attributes in a table.

An attribute is used to define the record and a record contains a set of attributes

Domain is the original set of atomic values used to model data.

Atomic value means each value in the domain is indivisible as far as the relational model is concerned.

A domain is a set of acceptable values that a column is allowed to contain.

Properties of a table

Name

No duplicate rows

Entities in columns are atomic

Entries from columns are from the same domain based on their data type, including (number, character, date, logical)

Operations combining different data types are disallowed

Each attribute has a distinct name

The sequence of columns is insignificant

The sequence of rows is insignificant

An entity is an object in the real world with an independent existance that can be differentiated from other objects.

Could be physical object (lecturer, student, car)

Could be conceptual (Course, job, position)

Entities can be classified on their strength. An entity is considered weak if its tables are existence dependent (it cannot exist without a relationship with another entity or its primary key is derived from the primary key of the parent entity)

An entity is considered strong if it can exist apart from all of its related entities

Kernels are strong entities

A table without a foreign key or a table that contains a foreign key that can contain nulls is a strong entity

Kinds of entities

Independent entities (kernals) – backbone of database. They are what other tables are based on. The’re primary key may be simple or composite, the primary key isnt a foreign key, they do not depend on another entity for their existence.

Dependent entities (derived entities), depend on other tables for their meaning. They’re used to connect two kernals together. They are said to be existence dependent on two or more tables. Many to many relationships become associative tables with at least two foreign keys. They may contain other attributes. The foreign key identifies each associated table.

Characteristic entities – provide more information about another table. They represent multivalued attributes, they describe other entities, they typically have a one-to-many relationship, the foreign key is used to further identify the characterised table

Types of attributes

Simple attributes – drawn from the atomic value domains; also known as single-valued attributes.

Composite attributes – those that consist of a hierarchy of attributes.

Multivalued attributes – have a set of values for each entity.

Derived attributes – contain values calculated from other attributes.

Keys

Key is an attribute or a group of attributes whose values can be used to uniquely identify an individual entity in an entity set.

Types of keys

Candidate key – simple or composite key that is unique and minimal.

Composite key – a composite key is composed of two or more attributes but it must be minimal.

Primary key – a candidate key that is selected by the database designer to be used as an identifying mechanism for the whole entity set. It must uniquely identify tuples in a table and not be null.

Secondary key – used strictly for retrieval purposes (can be composite)

Alternate key – all candidate keys not chosen as the primary key

Foreign key – is an attribute in a table that references the primary key in another table OR it can be null. Both foreign and primary key must be of the same type.

Nulls

A null is a special symbol, indepdenent of data type, meaning either unknown or inapplicable. It doesn’t mean zero or blank.

Features of null include: no data entry, not permitted in the primary key

Relationships

Relations are the glue holding tables together. They’re used to connect related information between tables.

A weak or non-identifying relationship exists if the primary key of the related entity doesn’t contain a primary key component of the parent entity.

Types of relationships

One to many – should be the norm. One department has many employees.

One-to-one- relationship of one entity to only one other entity. It should be rare. It could indicate that two entities belong in the same table.

Many-to-many – need to create a linking table, at least with the two primary keys.

A student can have many classes and a class can have many students.

Join table could be where an employee has a different start date for different projects.

Unary relationship (recursive) – is one in which a relationship exists between occurrences of the same entity. In this relationship, the primary and foreign keys are the same, but they represent two entities with different roles. For some entities, a separate column can be created that refers to the primary key.

Ternary relationship – involves many to many relationships between three tables.

Integrity rules

Domain integrity – restricts the values of attributes in the relation and is a constraint of the relational model.

Integrity constraints

Entity integrity – every table has a primary key which can’t contain null values.

Referential integrity – requires a foreign key must have a matching primary key or it must be null. This constraint is specified between two tables (parent and child).

Enterprise constraints – additional rules specified by users or database admins.

E.g. a class can have a maximum of 30 students, a teacher can teach a maximum of four classes per semester, etc.

Business rules – are obtained from users when gathering requirements. E.g. a teacher can teach many students, a course can be taught many times. Not all teachers teach classes.

Cardinality and connectivity

Cardinality describes the relationship between two data tables by expressing the min and max number of entity occurances associated with one occurrence of a related entity.

Connectivity is the relationship between two tables (one-to-one, one-to-many).

Relationship types

Identifying relationship will have a solid line (where the PK contains the FK).

Non-identifying relationship is indicated by a broken line (not containing the FK in the PK).

Optional relationships – the FK can be null or the parent table doesn’t need to have a corresponding child table occurrence.

Mandatory relationships – one entity occurrence requires a corresponding entity occurrence.

Database design try

Non normalised

|  |  |  |
| --- | --- | --- |
| Data name | Value Type | Value example |
|  |  |  |
| School |  |  |
| ID | Int |  |
| Type | Varchar(100) |  |
| Name | Varchar(255) |  |
| Years | TINYINT |  |
| Subjects | Varchar(50) |  |
| Sets | Varchar(20) |  |
| Levels | Varcar(20) |  |
| Teachers |  |  |
| Groups |  |  |
| Parents |  |  |
| Pupils |  |  |
|  |  |  |
| Pupil |  |  |
| ID | Int |  |
| First name | Varcar(40) |  |
| Last name | Varcar(40) |  |
| Enrol date | DATE |  |
| Graduate date | DATE |  |
| User image | VARCAR(255) |  |
| Create date | TIMESTAMP |  |
| Year |  |  |
| Set |  |  |
| Subjects |  |  |
| Classes enrolled |  |  |
| Behaviour rating | DECIMAL |  |
| Posts |  |  |
| Parent |  |  |
| Tutor |  |  |
|  |  |  |
| Parent |  |  |
| ID | INT |  |
| First name | Varcar(40) |  |
| Last name | Varcar(40) |  |
| Create date | TIMESTAMP |  |
| Child |  |  |
| Classes |  |  |
| Posts |  |  |
|  |  |  |
| Teacher |  |  |
| ID | INT |  |
| First name | Varcar(40) |  |
| Last name | Varcar(40) |  |
| Job title | VARCHAR(255) |  |
| Teacher role | VARCHAR(40) |  |
| Date created | TIMESTAMP |  |
| Classes |  |  |
| Child |  |  |
|  |  |  |
| Group |  |  |
| ID | INT |  |
| Class code | VARCAR(255) |  |
| Group name | VARCHAR(255) |  |
| Subject |  |  |
| Level |  |  |
| Set |  |  |
| Start date | DATETIME |  |
| End date | DATETIME |  |
| Date created | TIMESTAMP |  |
|  |  |  |
| Post |  |  |
| ID | INT |  |
| Post type | VARCHAR(10) |  |
| Post title | TINYTEXT |  |
| Post text | LONGTEXT |  |
| Post attachment | BOOLEAN |  |
| Post attachment name | VARCHAR(255) |  |
| Post attachment type | VARCHAR(40) |  |
| Post attachment url | VARCHAR(255) |  |
| Post time | TIMESTAMP |  |
| Author |  |  |
| Class |  |  |
| Topic |  |  |
| Notifications enabled | BOOLEAN |  |
| Scheduled | DATETIME |  |
|  |  |  |
| Task |  |  |
| ID | INT |  |
| title | TINYTEXT |  |
| Instructions | LONGTEXT |  |
| Due date | DATETIME |  |
| Issue date | DATETIME |  |
| Timestamp | TIMESTAMP |  |
| Author |  |  |
| Notification enabled | BOOLEAN |  |
| Post attachment | BOOLEAN |  |
| Post attachment name | VARCHAR(255) |  |
| Post attachment type | VARCHAR(40) |  |
| Post attachment url | VARCHAR(255) |  |
| Post time | TIMESTAMP |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Discussion |  |  |
| ID | Int |  |
| Title | TINYTEXT |  |
| Description | LONGTEXT |  |
| Timestamp | TIMESTAMP |  |
| Author |  |  |
| Class |  |  |
| Notification enabled |  |  |
| Post attachment | BOOLEAN |  |
| Post attachment name | VARCHAR(255) |  |
| Post attachment type | VARCHAR(40) |  |
| Post attachment url | VARCHAR(255) |  |
| Post time | TIMESTAMP |  |
|  |  |  |
|  |  |  |
| Reminder |  |  |
| ID |  |  |
| Title |  |  |
| Description |  |  |
| Reminder date |  |  |
| Timestamp |  |  |
| Author |  |  |
| Notification enabled |  |  |
| Post attachment | BOOLEAN |  |
| Post attachment name | VARCHAR(255) |  |
| Post attachment type | VARCHAR(40) |  |
| Post attachment url | VARCHAR(255) |  |
| Post time | TIMESTAMP |  |
|  |  |  |
| Resource |  |  |
| ID |  |  |
| Title |  |  |
| Description |  |  |
| Timestamp |  |  |
| Author |  |  |
| Notification enabled |  |  |
| Post attachment | BOOLEAN |  |
| Post attachment name | VARCHAR(255) |  |
| Post attachment type | VARCHAR(40) |  |
| Post attachment url | VARCHAR(255) |  |
| Post time | TIMESTAMP |  |
|  |  |  |
| Topic |  |  |
| ID |  |  |
| Title |  |  |
| Introduction |  |  |
| Timestamp |  |  |
| Author |  |  |
| Notification enabled |  |  |
| Post attachment | BOOLEAN |  |
| Post attachment name | VARCHAR(255) |  |
| Post attachment type | VARCHAR(40) |  |
| Post attachment url | VARCHAR(255) |  |
| Post time | TIMESTAMP |  |

1. Ryde Academy
   1. Levels 3, 4 and 5
      1. Years 9, 10, 11, 12, 13/14
         1. Studying English, Maths, Science
            1. Blue band

Set 1

Class

Teacher

Student

Parent

Topics

1. Sub-topic
   1. Title
   2. Text
2. Information
   1. Title
   2. Text
   3. Author
   4. Date created
   5. Date released
3. Reminders
4. Resources
5. Tasks
6. Discussions

* Ryde Academy
  + Teachers English, Maths, Science
    - Key state 3, 4 and 5
      * Contains years 9, 10, 11, 12, 13/14
        + Blue, purple or green band

Sets 1, 2, 3, 4