Database Models

|  |  |
| --- | --- |
| **Database Model** | **Description** |
| Hierarchical | One of the oldest database models, commonly used in mainframe computing.  Hierarchical with parent and child nodes. |
| Network | Hierarchical with owners and members.  Each member can have more than one owner. |
| Object-Oriented | Persistent storage of program objects such as class definitions.  Objects can survive past the end of program execution.  Impedance mismatch - data structures in DBMS are incompatible with the programming language’s data structures. |
| Graph | Uses a graph structure with nodes, edges and properties.  Treats the relationship between things as equally important to the things themselves. |
| Relational | More flexible than either hierarchical or network database models.  Relations (tables), tuples (rows) & attributes (columns). |

Relational Databases

|  |  |
| --- | --- |
| Table / Relation | Basic storage structure of a relational database.  Tables are two-dimensions. |
| Row / Tuple | Represents a collection of related values. |
| Column / Attribute | Contains values of the same data type. |

The data type of a column is represented by a **domain** of possible values.

The **degree** of a relational schema is equal to the number of attributes.

### Characteristics of Relations

A **relation** is defined as a set of tuples.

Elements of a set have **no order** among them.

Each tuple is an **ordered** list of values.

The order can change as long as the **correspondence** between attributes and values is maintained.

Values in tuples are **atomic**:

* **Multi-valued attributes** must be represented by **separate relations**.
* **Composite attributes** are represented by **simple** component attributes.

**NULL** values are used to represent the values of attributes that may be **unknown** or **may not apply** to a tuple.

### Constraints

**Constraints** are restrictions on the values that can be placed in a database state:

|  |  |
| --- | --- |
| **Implicit** Constraints | Constraints that are inherent to the data model. |
| Schema-based /  **Explicit** Constraints | Constraints expressed in the schemas of the data model with DDL. |
| Application-based /  **Semantic** Constraints | Constraints that cannot be expressed in the DDL.  e.g. business rules. |

### Key & Integrity Constraints

A **primary key** uniquely identifies a tuple.

The primary key is chosen from a set of **candidate keys**.

Multiple attributes can be combined to form a **composite primary key**.

A NULL value is not permitted as the primary key.

Key and entity constraints are enforced by the **DBMS**.

### Referential Integrity

**Referential integrity constraints** are specified between two relations. They maintain consistency among tuples in the two relations.

“Foreign keys can’t point to nuttin’.” - Ciarán Ingle, 2K18

A tuple in one relation that refers to another relation, must refer to an existing tuple in that relation.

A **foreign key** specifies a referential integrity constraint between two relations.