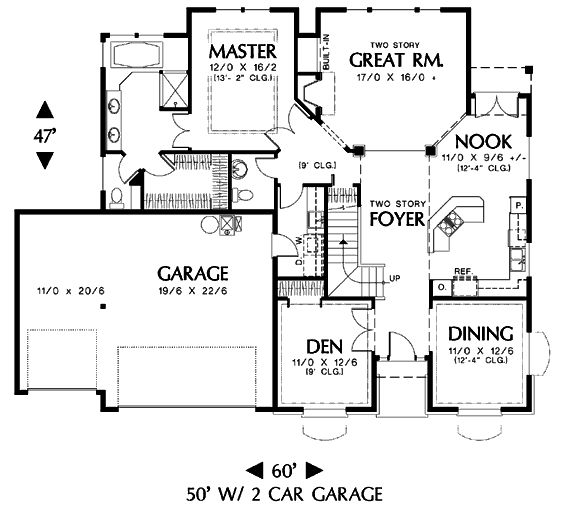
Data Model is a collection of conceptual tools for describing data, data relationships, data semantics and consistency constraint.

data model provides a way to describe the design of a database at the physical, logical and view levels.

**Conceptual:** describes **WHAT** the system contains.

**Logical:** describes **HOW** the system will be implemented, regardless of the DBMS.

**Physical:** describes **HOW** the system will be implemented using a **specific DBMS**.

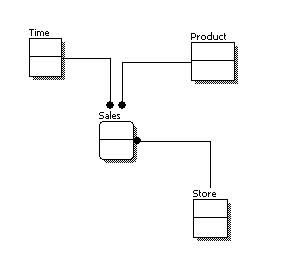
**Conceptual Data Model**

A conceptual data model identifies the **highest-level relationships** between the different entities. Features of conceptual data model include:

* Includes the important entities and the relationships among them.
* No attribute is specified.
* No primary key is specified.

The figure below is an example of a conceptual data model.

**Conceptual Data Model**



From the figure above, we can see that the only information shown via the conceptual data model is the entities that describe the data and the relationships between those entities. No other information is shown through the conceptual data model.

**Logical Data Model**

A logical data model describes the data in as much detail as possible, without regard to how they will be physical implemented in the database. Features of a logical data model include:

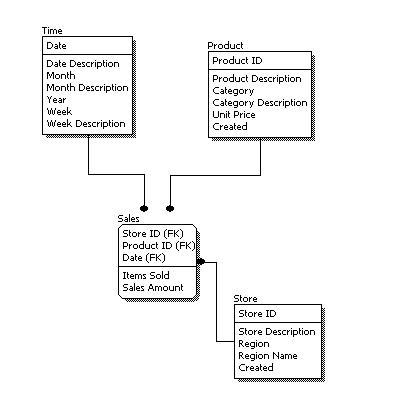
* Includes all entities and relationships among them.
* All attributes for each entity are specified.
* The primary key for each entity is specified.
* Foreign keys (keys identifying the relationship between different entities) are specified.
* Normalization occurs at this level.

The steps for designing the logical data model are as follows:

1. Specify primary keys for all entities.
2. Find the relationships between different entities.
3. Find all attributes for each entity.
4. Resolve many-to-many relationships.
5. Normalization.

The figure below is an example of a logical data model.

**Logical Data Model**



Comparing the logical data model shown above with the [conceptual data model](https://www.1keydata.com/datawarehousing/conceptual-data-model.html) diagram, we see the main differences between the two:

* In a logical data model, primary keys are present, whereas in a conceptual data model, no primary key is present.
* In a logical data model, all attributes are specified within an entity. No attributes are specified in a conceptual data model.
* Relationships between entities are specified using primary keys and foreign keys in a logical data model. In a conceptual data model, the relationships are simply stated, not specified, so we simply know that two entities are related, but we do not specify what attributes are used for this relationship.

**Physical Data Model**

Physical data model represents how the model will be built in the database. A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables. Features of a physical data model include:

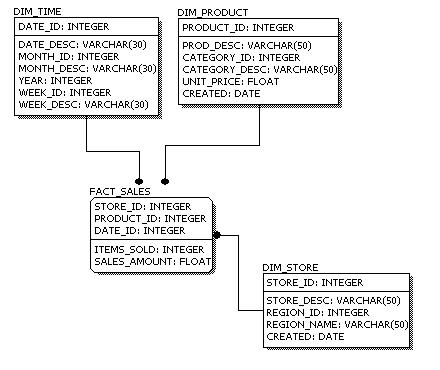
* Specification all tables and columns.
* Foreign keys are used to identify relationships between tables.
* Denormalization may occur based on user requirements.
* Physical considerations may cause the physical data model to be quite different from the logical data model.
* Physical data model will be different for different RDBMS. For example, data type for a column may be different between MySQL and SQL Server.

The steps for physical data model design are as follows:

1. Convert entities into tables.
2. Convert relationships into foreign keys.
3. Convert attributes into columns.
4. Modify the physical data model based on physical constraints / requirements.

The figure below is an example of a physical data model.

**Physical Data Model**



Comparing the physical data model shown above with the [logical data model](https://www.1keydata.com/datawarehousing/logical-data-model.html) diagram, we see the main differences between the two:

* Entity names are now table names.
* Attributes are now column names.
* Data type for each column is specified. Data types can be different depending on the actual database being used.
* The three levels of data modeling, [conceptual data model](https://www.1keydata.com/datawarehousing/conceptual-data-model.html), [logical data model](https://www.1keydata.com/datawarehousing/logical-data-model.html), and [physical data model](https://www.1keydata.com/datawarehousing/physical-data-model.html), were discussed in prior sections. Here we compare these three types of data models. The table below compares the different features:

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Conceptual | Logical | Physical |
| Entity Names | ✓ | ✓ |  |
| Entity Relationships | ✓ | ✓ |  |
| Attributes |  | ✓ |  |
| Primary Keys |  | ✓ | ✓ |
| Foreign Keys |  | ✓ | ✓ |
| Table Names |  |  | ✓ |
| Column Names |  |  | ✓ |
| Column Data Types |  |  | ✓ |

* Below we show the conceptual, logical, and physical versions of a single data model.

|  |  |  |
| --- | --- | --- |
| **Conceptual Model Design**  Conceptual Model Design | **Logical Model Design**  Logical Model Design | **Physical Model Design**  Physical Model Design |

* We can see that the complexity increases from conceptual to logical to physical. This is why we always first start with the conceptual data model (so we understand at high level what are the different entities in our data and how they relate to one another), then move on to the logical data model (so we understand the details of our data without worrying about how they will actually implemented), and finally the physical data model (so we know exactly how to implement our data model in the database of choice). In a data warehousing project, sometimes the conceptual data model and the logical data model are considered as a single deliverable.

