

Cymax

1. Foreign Key Vs Primary Key

Primary key uniquely identify a record in the table.

Foreign key is a field in the table that is primary key in another table.

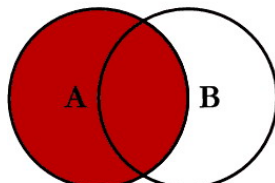
By default, Primary key is clustered index and data in the database table is physically organized in the sequence of clustered index.

Difference between Primary Key and Foreign Key

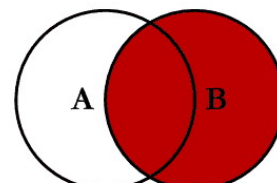
Primary Key	Foreign Key
Primary key uniquely identify a record in the table.	Foreign key is a field in the table that is primary key in another table.
Primary Key can't accept null values.	Foreign key can accept multiple null value.
By default, Primary key is clustered index and data in the database table is physically organized in the sequence of clustered index.	Foreign key do not automatically create an index, clustered or non-clustered. You can manually create an index on foreign key.
We can have only one Primary key in a table.	We can have more than one foreign key in a table.

2. Left, Right, Outer and Inner Joins?

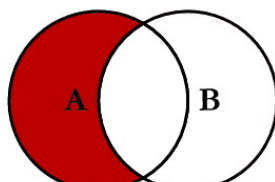
SQL JOINS



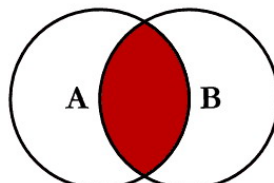
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



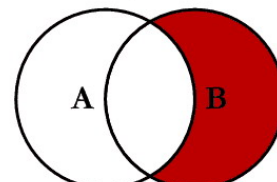
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



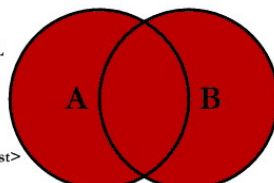
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



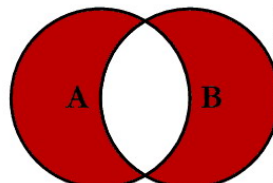
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

3. Difference Between SQL and NoSQL

Table: 某种特定类型数据的结构化清单。

Schema: 模式可以用来描述数据库中特定的表，也可以用来描述整个数据库（和其中表的关系）。

Column: 表中的一个字段。所有表都是由一个或多个列组成的。

行row: 表中的一个记录

1) Nature

MySQL: Relational database in nature.

NoSQL: Non-relational database in nature

2) Design

MySQL: Modeled based on the concept of 'table'

NoSQL: Modeled based on the concept of 'document'

3) Scalable

MySQL: Being relational in nature, can be tough task to scale big data

NoSQL: Easily scalable big data as compared to relational

4) Model

MySQL: Detailed database model needs to be in place before creation

NoSQL: No need to develop a detailed database model

5) Standardization

MySQL: SQL is standard language

NoSQL: Lack of a standard query language

6) Schema

MySQL: Schema is rigid

NoSQL: Dynamic schema is key benefit of NoSQL

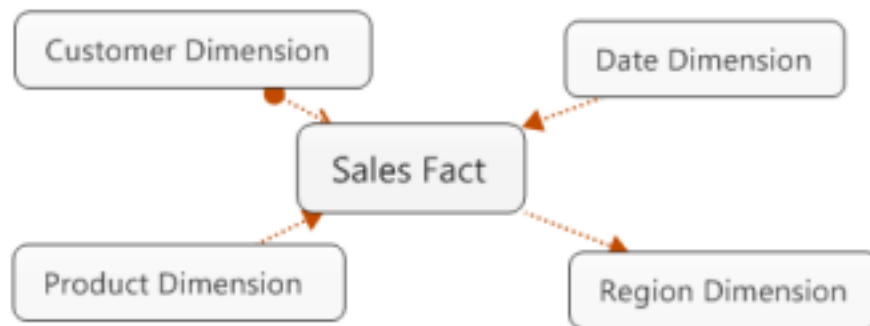
7) Flexibility

MySQL: Not so flexible design-wise, new column or field insertion affects design

NoSQL: New column or fields can be inserted without existing design

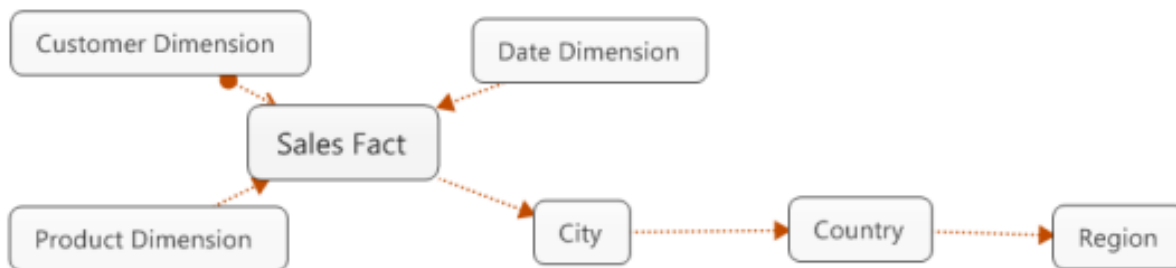
4. Star Schema and Snowflake Schema

1) Star Schema



The star schema is the simplest type of Data Warehouse schema. It is known as star schema as **its structure resembles a star**. In the Star schema, **the center of the star can have one fact tables and numbers of associated dimension tables**.

2) Snowflake Schema



A Snowflake Schema is **an extension of a Star Schema**, and it adds additional dimensions. It is called snowflake because its diagram resembles a Snowflake. The **dimension tables are normalized** which **splits data into additional tables**. In the following example, City is further normalized into an individual table.

Star Vs Snowflake Schema: Key Differences

Star Schema	Snow Flake Schema
Hierarchies for the dimensions are stored in the dimensional table.	Hierarchies are divided into separate tables.
It contains a fact table surrounded by dimension tables.	One fact table surrounded by dimension table which are in turn surrounded by dimension table
In a star schema, only single join creates the relationship between the fact table and any dimension tables.	A snowflake schema requires many joins to fetch the data.
Simple DB Design.	Very Complex DB Design.
Denormalized Data structure and query also run faster.	Normalized Data Structure.
High level of Data redundancy	Very low-level data redundancy
Single Dimension table contains aggregated data.	Data Split into different Dimension Tables.
Cube processing is faster.	Cube processing might be slow because of the complex join.
Offers higher performing queries using Star Join Query Optimization. Tables may be connected with multiple dimensions.	The Snow Flake Schema is represented by centralized fact table which unlikely connected with multiple dimensions.