

Relational databases and SQL

Tapti Palit

Relational databases

- Consists of tables
- Each table contains a primary key
 - Database will not allow insertion of two records with same primary key



The background features a faint watermark of a water tower and a bridge.

Products tbl		
ProductId	ProductName	ProductCount
1001	ABC	10

Customer tbl		
CustomerId	CustomerName	CustomerAddr
9001	XYZ	CDEF

Order tbl		
OrderId	CustomerId	ProductId
110001	9001	1001

Structured query language (SQL)

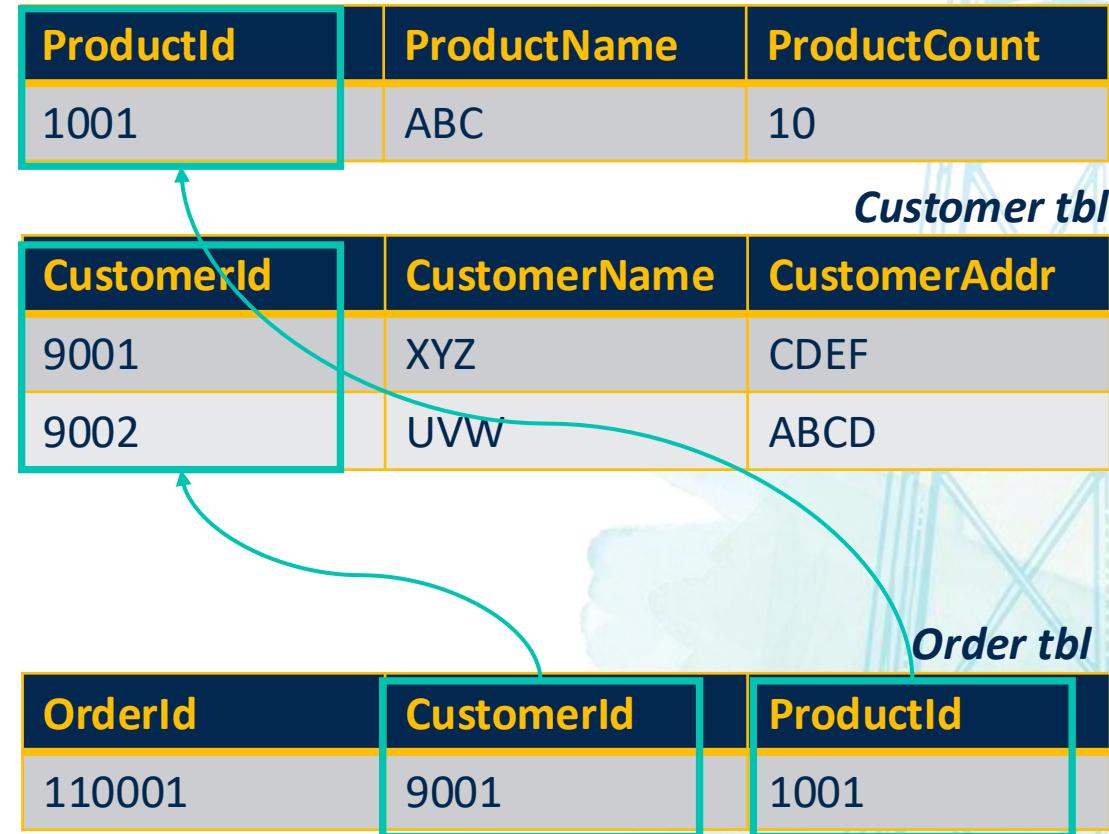
- SQL used to interface between application and database
- ```
CREATE TABLE Products (
 ProductId INT PRIMARY KEY,
 ProductName VARCHAR(100) NOT NULL,
 ProductCount INT NOT NULL);
```
- ```
INSERT INTO Products (ProductId, ProductName,
ProductCount) VALUES (1001, 'ABC', 10);
```

Structured query language (SQL)

- `SELECT * FROM Products; // returns all rows`
- `SELECT ProductId, ProductName FROM Products; // returns only the two columns`
- `SELECT * FROM Products WHERE ProductName = "XYZ"; // returns only the rows where product name is XYZ`

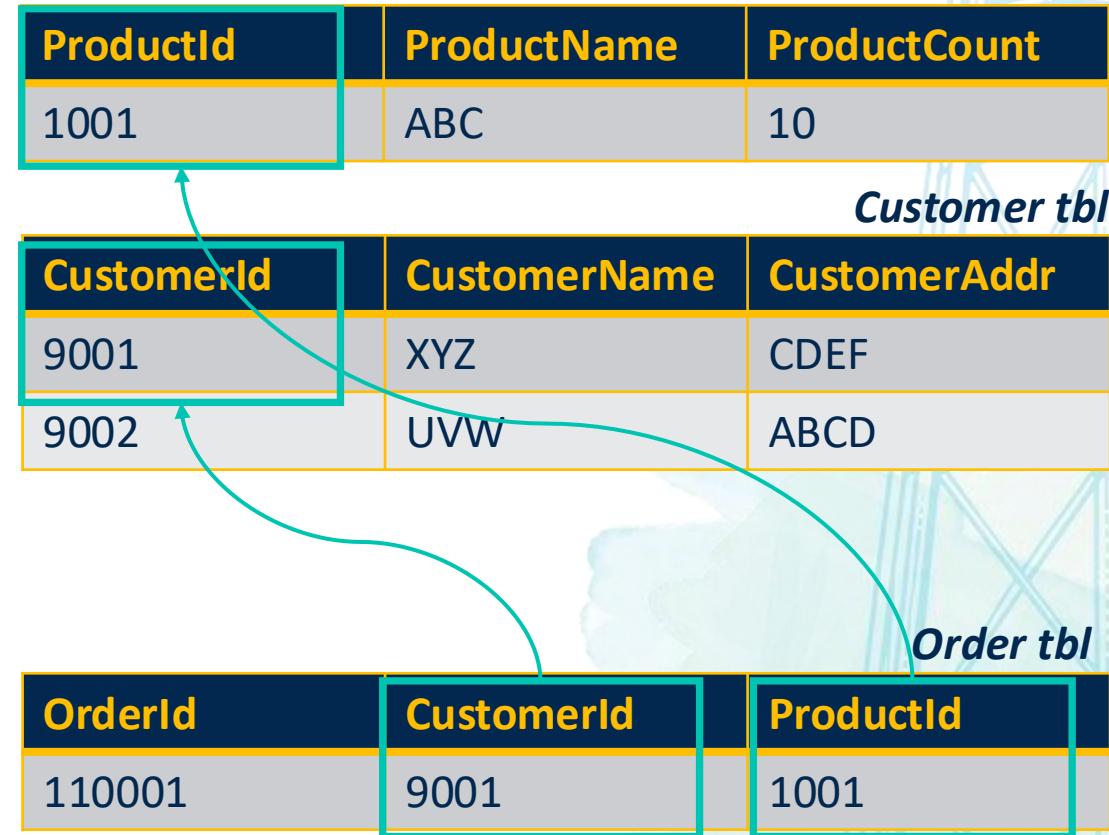
Foreign keys

- Relational databases maintain relations through foreign keys
- Foreign keys **must refer** to primary keys of other tables
 - Enforce referential integrity
- A table can contain one or more foreign keys



SQL joins

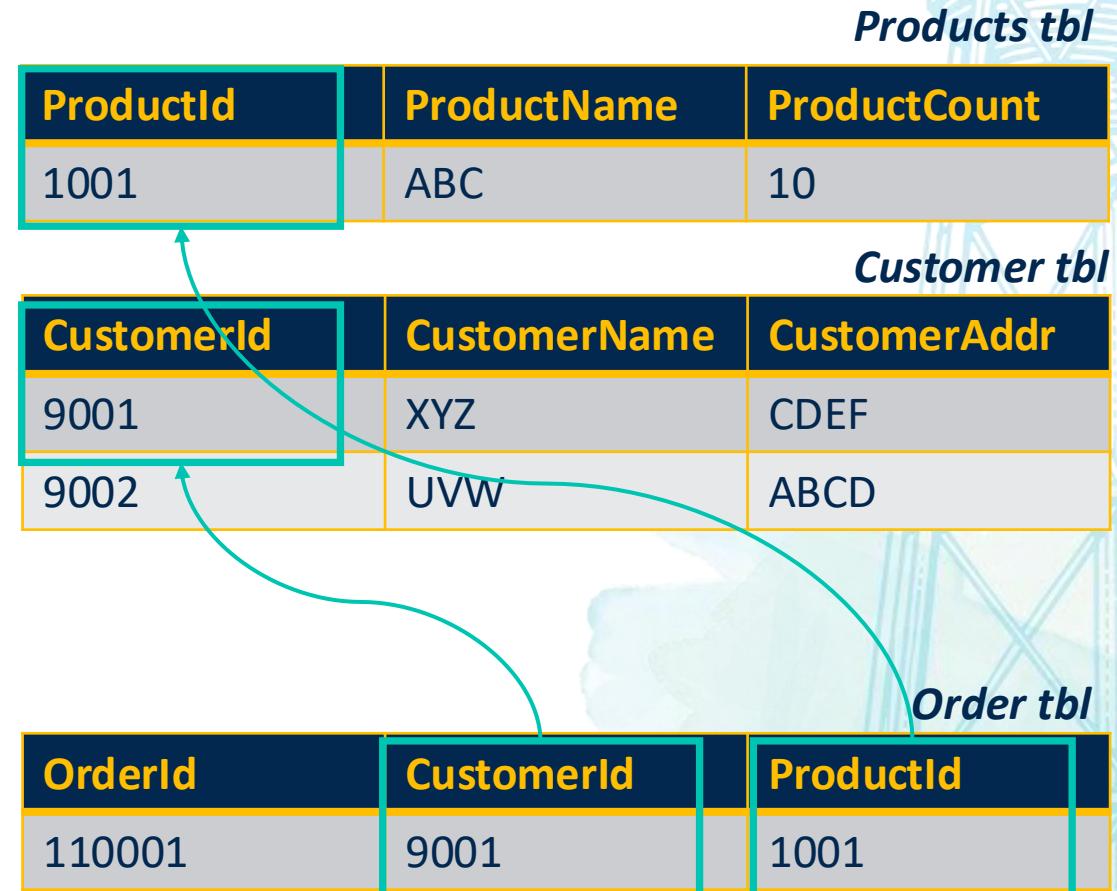
- Allows table joins
- For e.g. find order details for shipping, including product name, customer name, and address



SQL joins

SELECT

```
    o.OrderId,  
    p.ProductName,  
    c.CustomerName,  
    c.CustomerAddr  
  
FROM Orders o  
JOIN Products p  
    ON o.ProductId = p.ProductId  
JOIN Customers c  
    ON o.CustomerId =  
c.CustomerId;
```



Java database connectivity interface (JDBC)

- Interface for talking to SQL databases from Java
- PreparedStatement API to create a new query
- `pstmt.executeUpdate()` to execute the update

```
Post p = new Post(101, "Author1");
Connection connection =
DriverManager.getConnection("jdbc:sqlite:" +
dbPath);
```

```
String sql = "INSERT INTO Post (cid,
author_handle) " + "VALUES (?, ?)";
```

```
try {
PreparedStatement pstmt =
connection.prepareStatement(sql));
pstmt.setString(1, post.getCid());
pstmt.setString(2,
post.getAuthorHandle());
pstmt.executeUpdate();
} catch (Exception e) {--- snip }
```

Java database connectivity interface (JDBC)

- `stmt.executeUpdate()` to execute the the select query
- `ResultSet API` to retrieve query results

```
String cid = "101";
String sql = "SELECT cid, author_handle WHERE cid
= ?";

try {
    PreparedStatement stmt =
connection.prepareStatement(sql);
    stmt.setString(1, cid);

    try (ResultSet rs = stmt.executeQuery()) {
        if (rs.next()) {
            Post post = new Post(
                rs.getString("cid");
                rs.getString("author_handle"),
            );
            return post;
        }
    } catch(Exception e) {}
} catch (Exception e2) {}
```