

# SQL VIEWS

## The VIEW

- A “VIEW” is an empty shell of a table definition
- The view contains no data until it is queried
- Can be considered as a “Virtual Table”
- Each time the view is queried, the underlying query that populates the view is re-executed

# SQL VIEWS

## Why use VIEWS?

### 1. Security, maintaining confidentiality

The view can prevent unauthorized users from seeing data that must be kept secure or confidential

### 2. Complexity

The view can hide complex SQL from unauthorized or inexperienced users

# SQL VIEWS

First case:

**Base Table:**

Employees(EmpID, Lastname, Firstname, Salary, HireDate)

**View:**

Employees(EmpID, Lastname, Firstname, HireDate)

# SQL VIEWS

## Second case:

### Base Query

```
CREATE VIEW "alanparadise/nw"."TopEmployeesOrders" AS
  SELECT LastName, Firstname, sum(unitprice * quantity) as "Total Sales"
    from "alanparadise/nw"."employees" E JOIN
      "alanparadise/nw"."orders" O ON E.employeeid = O.employeeid JOIN
        "alanparadise/nw"."orderdetails" D ON O.orderid = D.orderid
  GROUP BY LastName, FirstName
  Order By 3 desc LIMIT 5;
```

### View

```
Select * from TopEmployeeOrders;
```

# SQL VIEWS

## CREATING a VIEW

```
CREATE VIEW <view name> AS  
    SELECT <col1>, <col2>, <col3>  
    FROM <table1>  
    WHERE <condition>
```

# SQL VIEWS

```
CREATE VIEW "alanparadise/nw"."TopEmployeesOrders" AS
  SELECT LastName, Firstname, sum(unitprice * quantity) as "Total Sales"
    from "alanparadise/nw"."employees" E
      JOIN
        "alanparadise/nw"."orders" O ON E.employeeid = O.employeeid
      JOIN
        "alanparadise/nw"."orderdetails" D ON O.orderid = D.orderid
  GROUP BY LastName, FirstName
  Order By 3 desc LIMIT 5;
```

## Querying the View:

```
Select * from "alanparadise/nw"."TopEmployeesOrders";
```