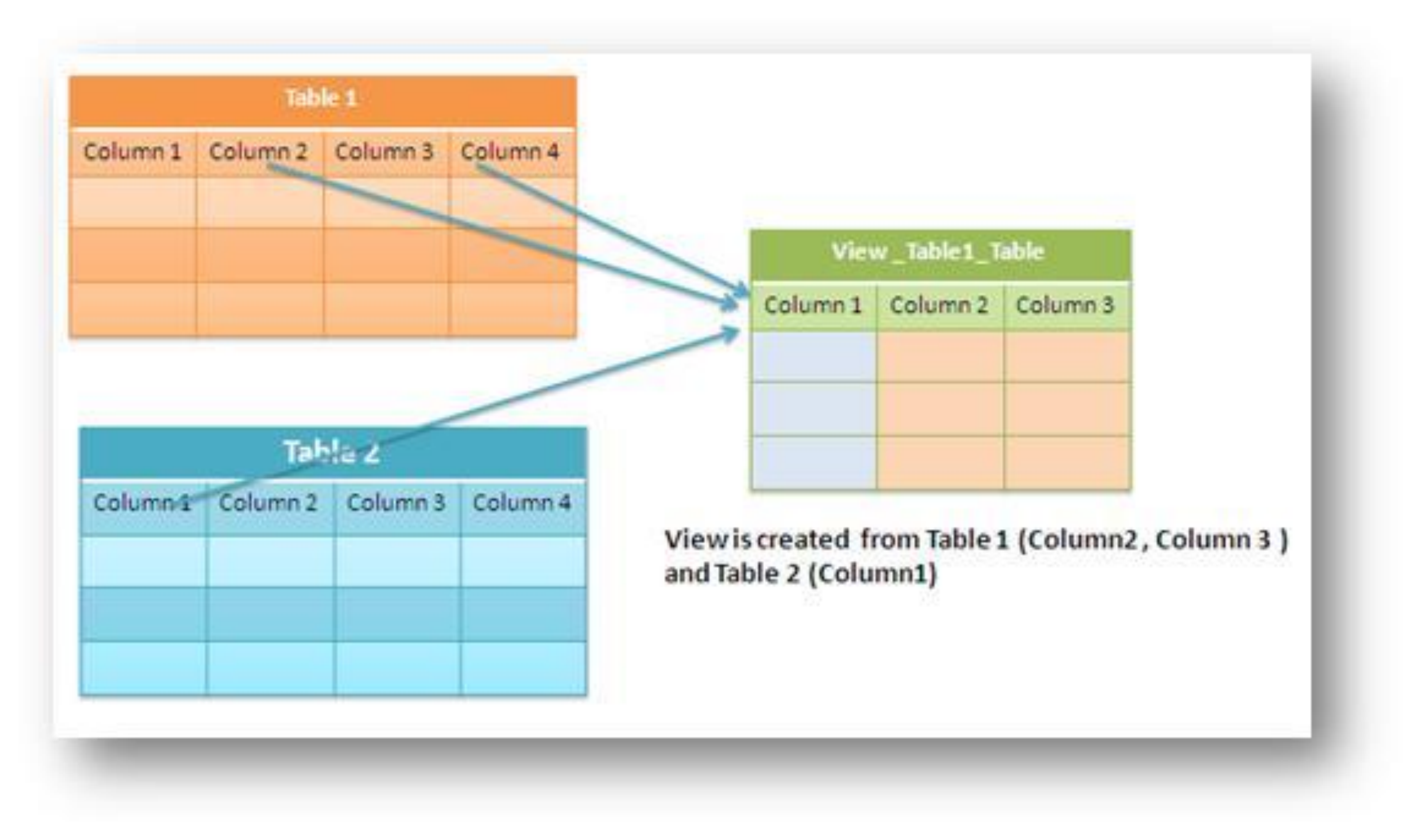
INFORMATION MANAGEMENT LAB (LAB 8:VIEWS)

Views

- In some cases, it is not desirable for all users to see the entire logical model (that is, all the actual relations stored in the database.)
- Consider a person who needs to know a customer’s loan number but has no need to see the loan amount. This person should see a relation described, in SQL, by

```
(select customer_name, loan_number
from borrower, loan
where borrower.Loan_number = loan.Loan_number )
```

- **<u>A view is a "virtual table". It is not like a simple table, but is a virtual table which contains columns and data from different tables (may be one or more tables)</u>**



View is created from Table 1 (Column2 , Column 3) and Table 2 (Column1)

View definition

- An SQL View is a specific representation of data from one or more tables.
- The tables referred in the views are known as base tables.
- View is virtual relation that does not necessarily actually exist in the database but is produced upon request, at time of request.
- Like a table, a **view** consists of a set of named columns and rows of data. Unless indexed, a **view** does not exist as a **stored** set of data values in a **database**.

Advantages of views

- **Security** each user can be given permission to access the database only through a small set of views that contain the specific data the user is authorized to see
- **Query simplicity** A view can draw data from several different tables and present it as a single table, turning multi-table queries into single-table queries against the view.
- **Structural simplicity** views can give a user a "personalized" view of the database structure, presenting the database as a set of virtual tables that make sense for that user.
- **Consistency** A view can present a consistent, unchanged image of the structure of the database, even if the source tables are split, restructured, or renamed.
- **Data integrity** if data is accessed and entered through a view, the dbms can automatically check the data to ensure that it meets the specified integrity constraints.

Create views

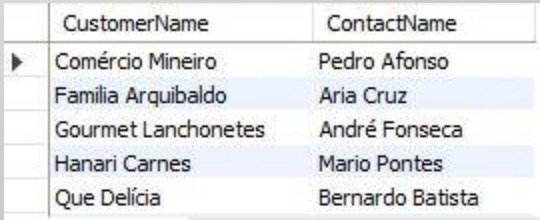
Syntax:

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Example

Query: Creates a view that shows all customers from brazil:

```
CREATE VIEW brazil_customers AS
SELECT customername, contactname
FROM customers
WHERE country = 'brazil';
```

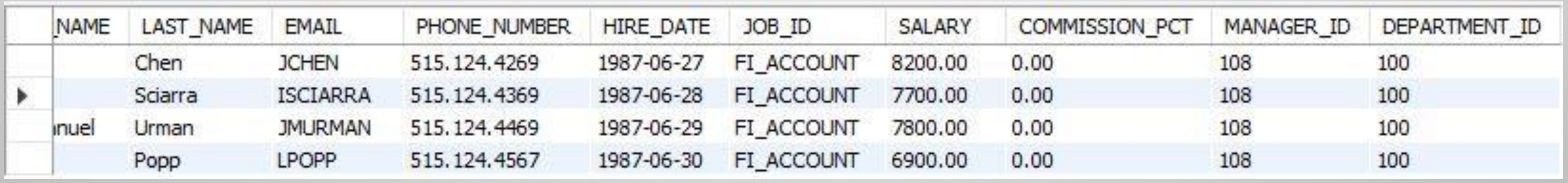


CustomerName	ContactName
Comércio Mineiro	Pedro Afonso
Familia Arquibaldo	Aria Cruz
Gourmet Lanchonetes	André Fonseca
Hanari Carnes	Mario Pontes
Que Delícia	Bernardo Batista

Creating view from a single table:

- Write a query to create a view for those employees belong to the department ID as 100 and display the view.

```
Create view department_100 as
Select * from employees
Where department_id=100;
Select * from department_100;
```



	NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
		Chen	JCHEN	515.124.4269	1987-06-27	FI_ACCOUNT	8200.00	0.00	108	100
▸		Sciarra	ISCIARRA	515.124.4369	1987-06-28	FI_ACCOUNT	7700.00	0.00	108	100
	nuel	Urman	JMURMAN	515.124.4469	1987-06-29	FI_ACCOUNT	7800.00	0.00	108	100
		Popp	LPOPP	515.124.4567	1987-06-30	FI_ACCOUNT	6900.00	0.00	108	100

Creating view from multiple tables:

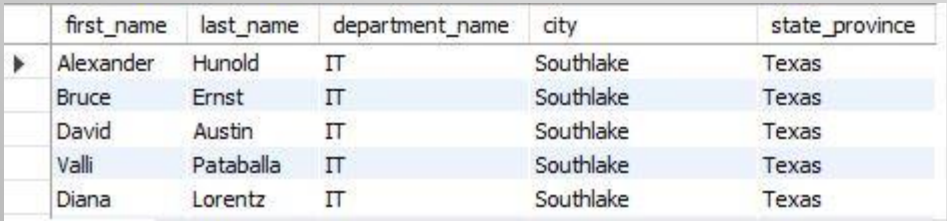
- Write a query to create the view of first and last name, department, city, and state province for each employee

```
CREATE VIEW on_join    as

SELECT e.first_name,e.last_name, d.department_name,
l.city,l.state_province

FROM locations l join departments d using (location_id) join employees e
using (department_id);

Select * from on_join;
```



first_name	last_name	department_name	city	state_province
Alexander	Hunold	IT	Southlake	Texas
Bruce	Ernst	IT	Southlake	Texas
David	Austin	IT	Southlake	Texas
Valli	Pataballa	IT	Southlake	Texas
Diana	Lorentz	IT	Southlake	Texas

Use VIEW as a table in another view

- Write a query in SQL to create a view and display those employees who contain a letter z to their first name and also display their last name, department, city, and state province.

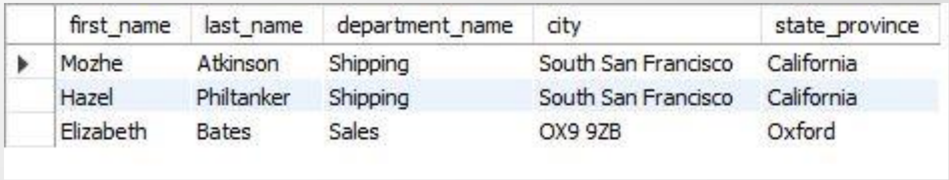
  Create view search_on_join as

  Select * from on_join

  Where first_name like '%z%’;

  Select * from search_on_join;

  Note: View on_join is used with the select statement for the abovementioned query.



	first_name	last_name	department_name	city	state_province
▶	Mozhe	Atkinson	Shipping	South San Francisco	California
	Hazel	Philtanker	Shipping	South San Francisco	California
	Elizabeth	Bates	Sales	OX9 9ZB	Oxford

Update views

Conditions needed to be satisfied to update a view.

- The select statement which is used to create the view should not include group by clause or order by clause.
- The select statement should not have the distinct keyword.
- The view should have all not null values.
- The view should not be created using nested queries or complex queries.
- The view should be created from a single table. If the view is created using multiple tables then we will not be allowed to update the view.

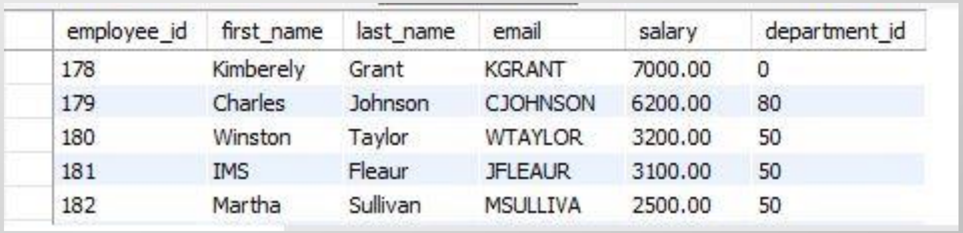
Update views example

- Write a query to create a view for all employees with columns employee_id, name, email, salary and department ID.

```
Create view employee_updated as
select e.Employee_id, e.First_name, e.Last_name, e.Email, e.Salary, e.Department_id
from employees e;
select * from employee_updated;
```

- Now UPDATE the employee first name as **your first name** whose employee_id is '181’

```
update employee_updated
set first_name='IMS’
where employee_id='181';
```



employee_id	first_name	last_name	email	salary	department_id
178	Kimberely	Grant	KGRANT	7000.00	0
179	Charles	Johnson	CJOHNSON	6200.00	80
180	Winston	Taylor	WTAYLOR	3200.00	50
181	IMS	Fleaur	JFLEAUR	3100.00	50
182	Martha	Sullivan	MSULLIVA	2500.00	50

Delete views

- If a created view is not needed then we can delete it.
- We can delete or drop a view using the drop statement.
- Example:

```
Drop view on_join;
```

THANK YOU..