

Kosta Nikopoulos
March 22nd, 2021

Lab 9 – Database Views

1. – Creating a single view

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane displays a tree of database objects for the 'xe-kosta2' connection, including tables like CAR, CUSTOMER, MECHANIC, PARTS, and PARTSUSED. The main workspace is divided into a 'Worksheet' and a 'Query Builder'. The 'Worksheet' contains the following SQL script:

```
--Lab 9
--Database Views
--Kosta Nikopoulos

--1. Create a VIEW that queries a table having at least 5 columns.
CREATE OR REPLACE VIEW vw_parts("Part_ID", "Part Number", "Description", "Buy Price", "Sell Price")
AS SELECT
    parts_id, part_number, description, purchase_price, retail_price
FROM parts
WITH READ ONLY;
```

Below the script, the 'Script Output' pane shows the message: 'View VW_PARTS created.' and a completion time of 0.089 seconds.

--Parts View

The screenshot shows the Oracle SQL Developer interface with the 'VW_PARTS' view selected. The 'Data' tab is active, displaying a table with 6 rows and 6 columns. The columns are: Part_ID, Part Number, Description, Buy Price, and Sell Price. The data is as follows:

	Part_ID	Part Number	Description	Buy Price	Sell Price
1	1	10010	Steel Doors	95	395
2	2	20020	Reinforced Windows	250	275
3	3	30030	Gauges	400	600
4	4	40040	Exhaust	150	200
5	5	50050	Sensors	50	100
6	6	60060	Breaking System	250	550

2. – Creating a view with 2 different tables

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane lists the database 'xe-kosta2'. The 'Tables (Filtered)' list includes CAR, CUSTOMER, MECHANIC, PARTS, PARTSUSED, SALES_INVOICE, SALESPERSON, SERVICE, SERVICEMECHANIC, and SERVICETICKET. The 'Views' list is empty. The 'Script Output' pane at the bottom shows the execution results of the SQL script.

```
--Lab 9
--Database Views
--Kosta Nikopoulos

--1. Create a VIEW that queries a table having at least 5 columns.
CREATE OR REPLACE VIEW vw_parts("Part_ID","Part Number", "Description", "Buy Price", "Sell Price")
AS SELECT
    parts_id,part_number, description,purchase_price,retail_price
FROM parts
WITH READ ONLY;

--2. Create a VIEW that read data from two different tables of your choice.
CREATE OR REPLACE VIEW vw_car_invoice
AS SELECT
    c.serial_number, c.car_id, si.invoice_number, si.si_date
FROM Car c, sales_invoice si
WHERE c.car_id = si.car_id
WITH READ ONLY;
```

Script Output x

Task completed in 0.04 seconds

View VW_PARTS created.

View VW_CAR_INVOICE created.

--Car Invoice View

The screenshot shows the Oracle SQL Developer interface with the 'VW_CAR_INVOICE' view selected. The 'Data' tab is active, displaying the following data:

	SERIAL_NUMBER	CAR_ID	INVOICE_NUMBER	SI_DATE
1	21547	117	50010	06-08-01
2	67347	138	60020	11-07-09
3	35245	215	70030	15-11-21
4	1279	99	80040	12-10-24
5	78546	101	90050	09-09-09
6	21547	777	10001	18-05-11

3. –Creating a view with 3 different tables

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane shows the 'xe-kosta2' connection. The 'Tables (Filtered)' pane shows a list of tables including CAR, CUSTOMER, MECHANIC, PARTS, PARTSUSED, SALES_INVOICE, SALESPERSON, SERVICE, SERVICEMECHANIC, and SERVICE_TICKET. The 'Worksheet' pane contains the following SQL script:

```
CREATE OR REPLACE VIEW vw_parts("Part_ID", "Part Number", "Description", "Buy Price", "Sell Price")
AS SELECT
    parts_id, part_number, description, purchase_price, retail_price
FROM parts
WITH READ ONLY;

--2. Create a VIEW that read data from two different tables of your choice.
CREATE OR REPLACE VIEW vw_car_invoice
AS SELECT
    c.serial_number, c.car_id, si.invoice_number, si.si_date
FROM Car c, sales_invoice si
WHERE c.car_id = si.car_id
WITH READ ONLY;

--3. Create a VIEW that read data from three different tables of your choice. The tables to be queried must be related through a parent-child relationship.
CREATE OR REPLACE VIEW vw_service_mech
AS SELECT
    m.first_name, m.last_name, s.service_id, sm.mechanic_id, sm.comments
FROM mechanic m, service s, servicesmechanic sm
WHERE m.mechanic_id = sm.mechanic_id
AND s.service_id = sm.service_id
WITH READ ONLY;

--4. Write DROP statements that delete the views you created in the previous steps.
DROP VIEW vw_parts;
DROP VIEW vw_car_invoice;
DROP VIEW vw_service_mech;
```

The 'Script Output' pane shows the following messages:

```
View VW_PARTS created.
View VW_CAR_INVOICE created.
View VW_SERVICE_MECH created.
```

--Service Mech View

The screenshot shows the Oracle SQL Developer interface with the 'VW_SERVICE_MECH' view selected. The 'Columns' pane shows the columns: FIRST_NAME, LAST_NAME, SERVICE_ID, MECHANIC_ID, and COMMENTS. The 'Data' pane shows the following data:

FIRST_NAME	LAST_NAME	SERVICE_ID	MECHANIC_ID	COMMENTS
1 Arthur	Reed	11		1 Oil was Changed
2 Niko	Reed	22		2 Windows were changed
3 Micheal	Micheal	33		3 Breaks were replaced
4 Jonny	Deep	44		4 Tires were changed
5 Jimmy	Michels	55		5 Exhaust was changed
6 Jack	Creed	66		6 ScheduledMaintenance

4. –Dropping the views

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane shows the 'xe-kosta2' connection. The 'Tables (Filtered)' pane shows a list of tables including VW_CAR_INVOICE, VW_PARTS, and VW_SERVICE_MECH. The 'Worksheet' pane contains the following SQL script:

```
CREATE OR REPLACE VIEW vw_parts("Part_ID", "Part Number", "Description", "Buy Price", "Sell Price")
AS SELECT
    parts_id, part_number, description, purchase_price, retail_price
FROM parts
WITH READ ONLY;

--2. Create a VIEW that read data from two different tables of your choice.
CREATE OR REPLACE VIEW vw_car_invoice
AS SELECT
    c.serial_number, c.car_id, si.invoice_number, si.si_date
FROM Car c, sales_invoice si
WHERE c.car_id = si.car_id
WITH READ ONLY;

--3. Create a VIEW that read data from three different tables of your choice. The tables to be queried must be related through a parent-child relationship.
CREATE OR REPLACE VIEW vw_service_mech
AS SELECT
    m.first_name, m.last_name, s.service_id, sm.mechanic_id, sm.comments
FROM mechanic m, service s, servicesmechanic sm
WHERE m.mechanic_id = sm.mechanic_id
AND s.service_id = sm.service_id
WITH READ ONLY;

--4. Write DROP statements that delete the views you created in the previous steps.
DROP VIEW vw_parts;
DROP VIEW vw_car_invoice;
DROP VIEW vw_service_mech;
```

The 'Script Output' pane shows the following messages:

```
View VW_PARTS dropped.
View VW_CAR_INVOICE dropped.
View VW_SERVICE_MECH dropped.
```