[www.oracle.com/academy](http://www.oracle.com/academy)

# Database Programming with SQL 15-1: Creating Views

Practice Solutions

## Vocabulary

Directions: Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| View | A subset of data from one or more tables that is generated  from a query and stored as a virtual table |
| VIEW\_NAME | Name of view |
| FORCE | Creates a view regardless of whether or not the base tables exist |
| Simple view | Derives data from one table, no functions or groups, performs  DML operations through the view |
| NOFORCE | Creates the view only if the base table exists |
| CREATE VIEW | Statement used to create a new view |
| Alias | Specifies a name for each expression selected by the view’s query |
| Subquery | A complete SELECT statement |
| Complex view | Derives data from one or more tables, contains functions or groups of data, and does not always allow DML operations  through the view |
| REPLACE | Re-creates the view if it already exists |

## Try It / Solve It

1. What are three uses for a view from a DBA’s perspective?

### Solution:

Any 3 from the following:

* + To restrict access to the data because the view can display selective columns from the table.
  + To reduce the complexity of executing queries based on more complicated SELECT statements.
  + To retrieve data from several tables, providing data independence for users. Users can view the same data in different ways.
  + To provide groups of users with access to data according to their particular permis- sions or criteria.

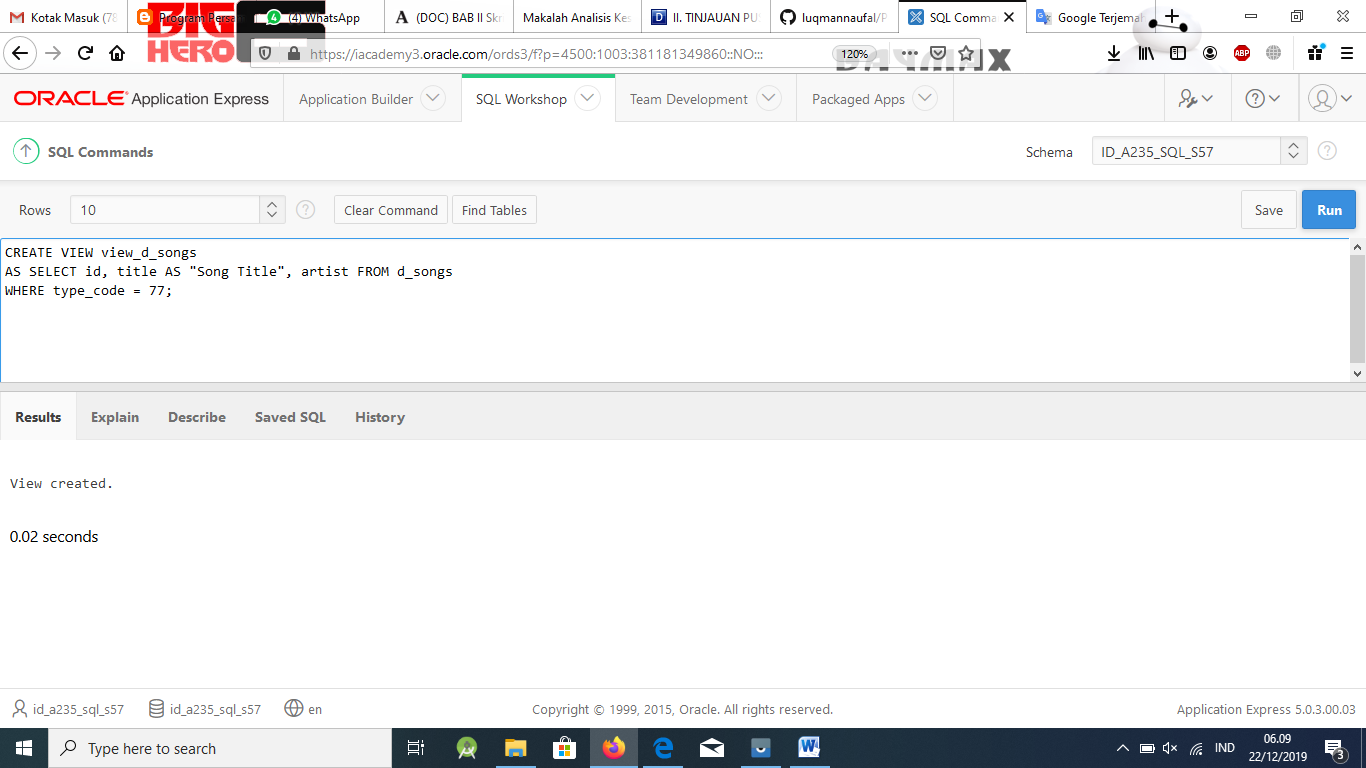
1. Create a simple view called view\_d\_songs that contains the ID, title and artist from the DJs on Demand table for each “New Age” type code. In the subquery, use the alias “Song Title” for the title column.

### Solution:

CREATE VIEW view\_d\_songs

AS SELECT id, title AS "Song Title", artist FROM d\_songs

WHERE type\_code = 77;

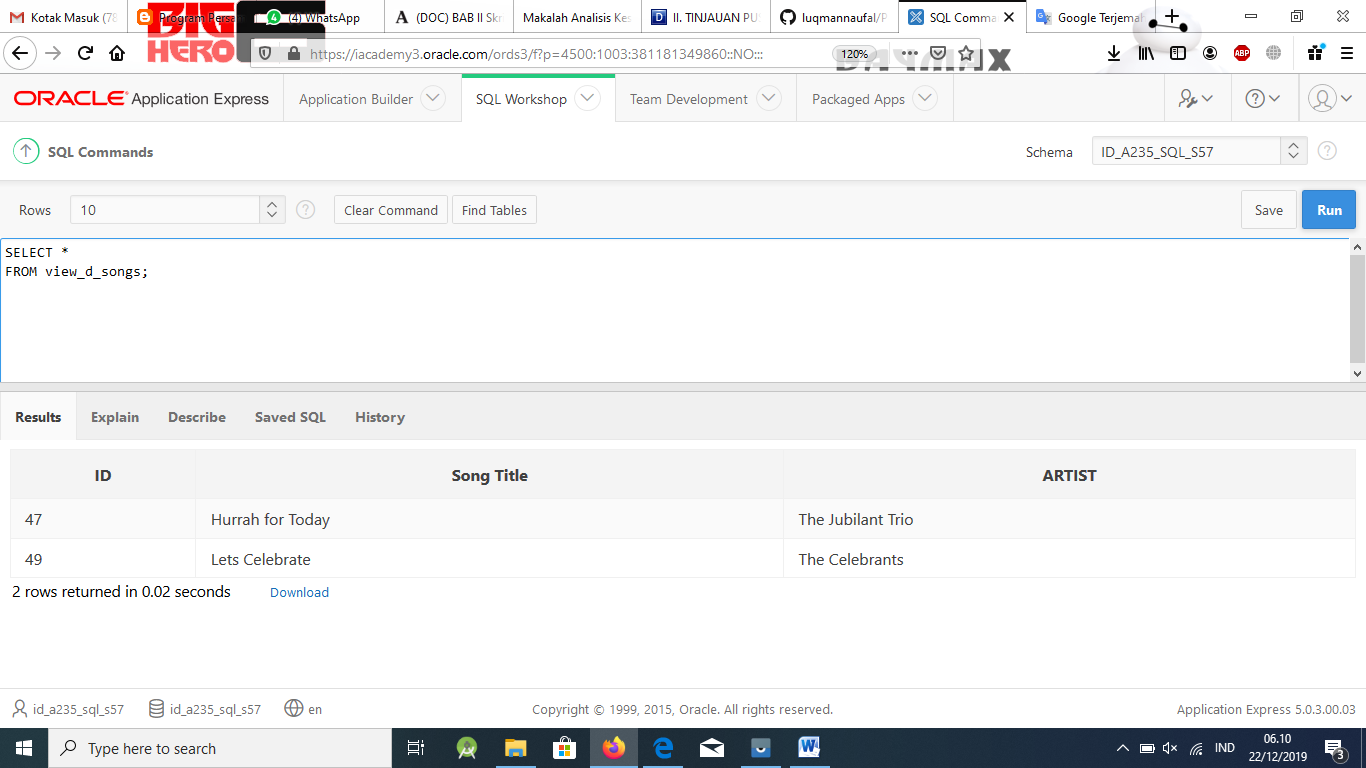


1. SELECT \* FROM view\_d\_songs. What was returned?

### Solution:

SELECT \*

FROM view\_d\_songs;



1. REPLACE view\_d\_songs. Add type\_code to the column list. Use aliases for all columns.

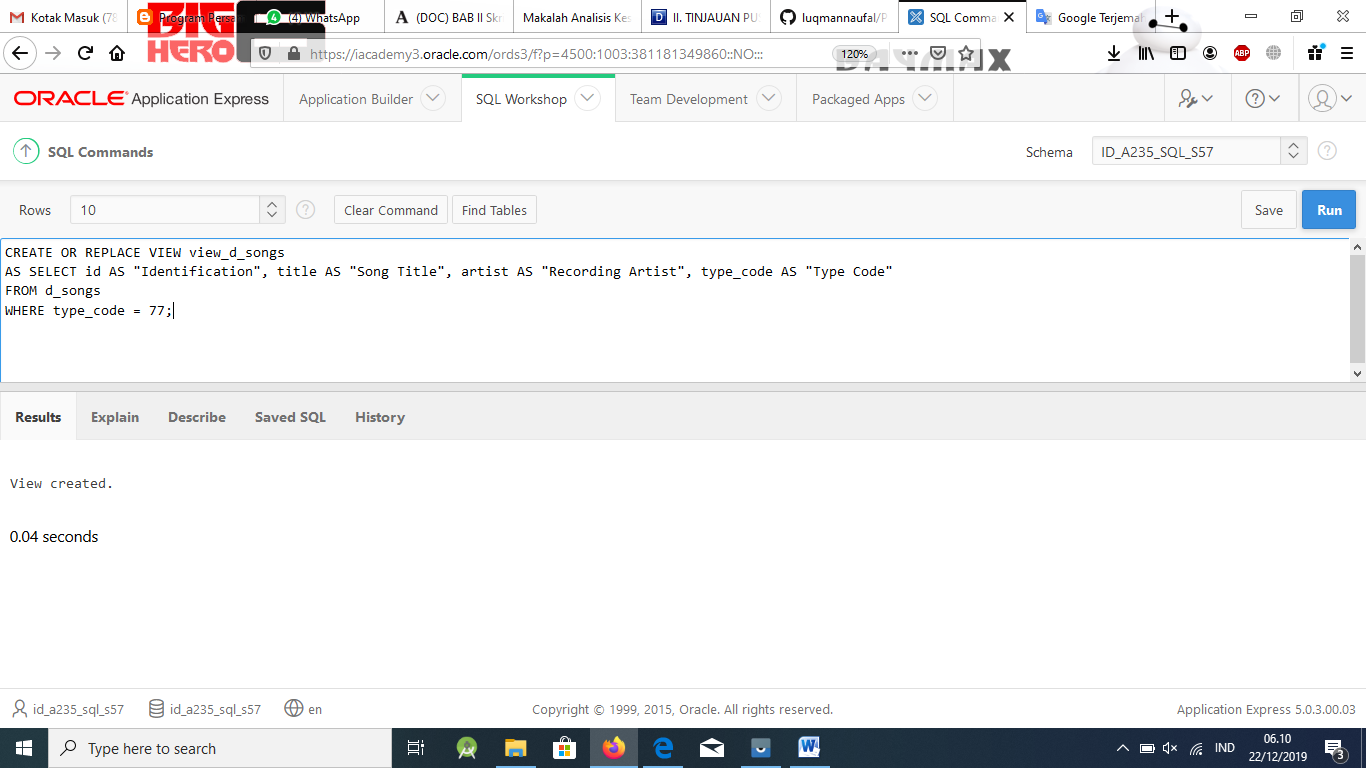
### Solution:

CREATE OR REPLACE VIEW view\_d\_songs

AS SELECT id AS "Identification", title AS "Song Title", artist AS "Recording Artist", type\_code AS "Type Code"

FROM d\_songs

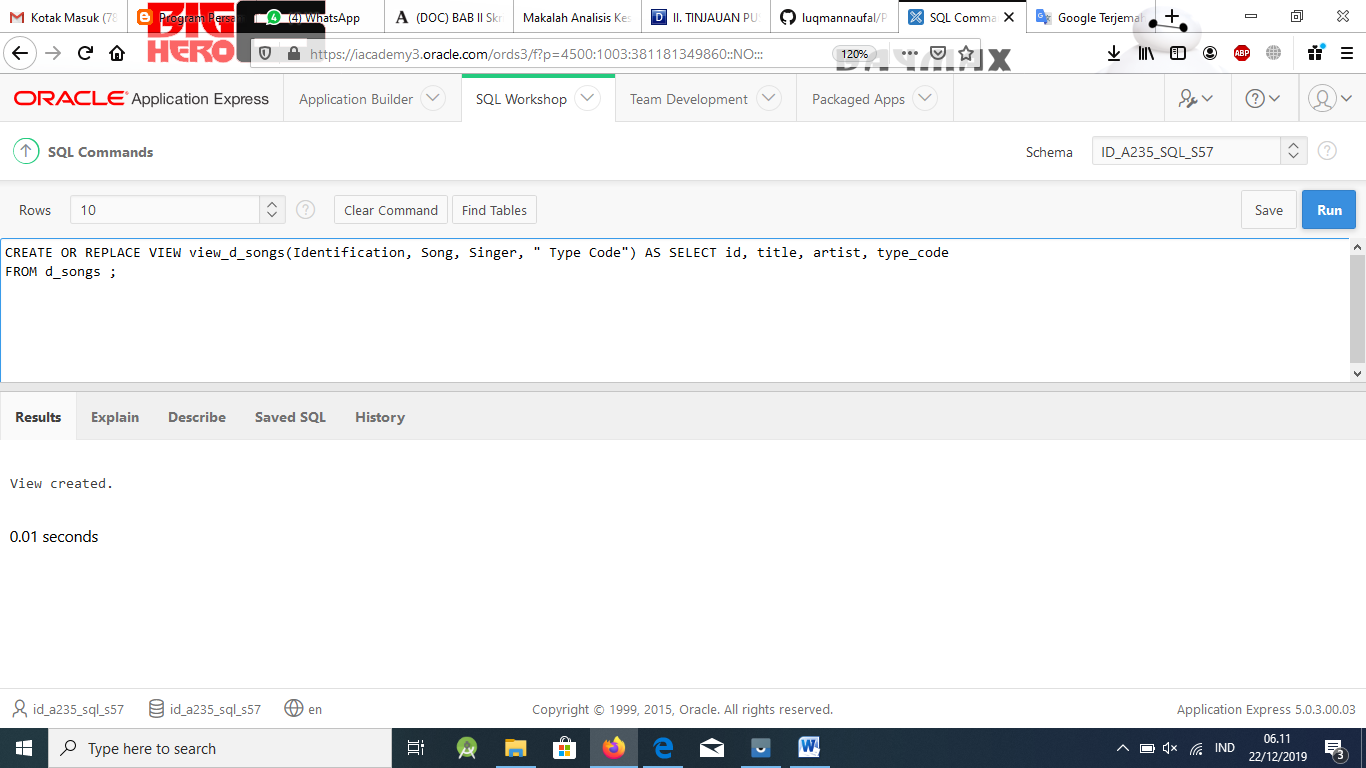
WHERE type\_code = 77;



Or use alias after the CREATE statement as shown.

CREATE OR REPLACE VIEW view\_d\_songs(Identification, Song, Singer, " Type Code") AS SELECT id, title, artist, type\_code

FROM d\_songs ;



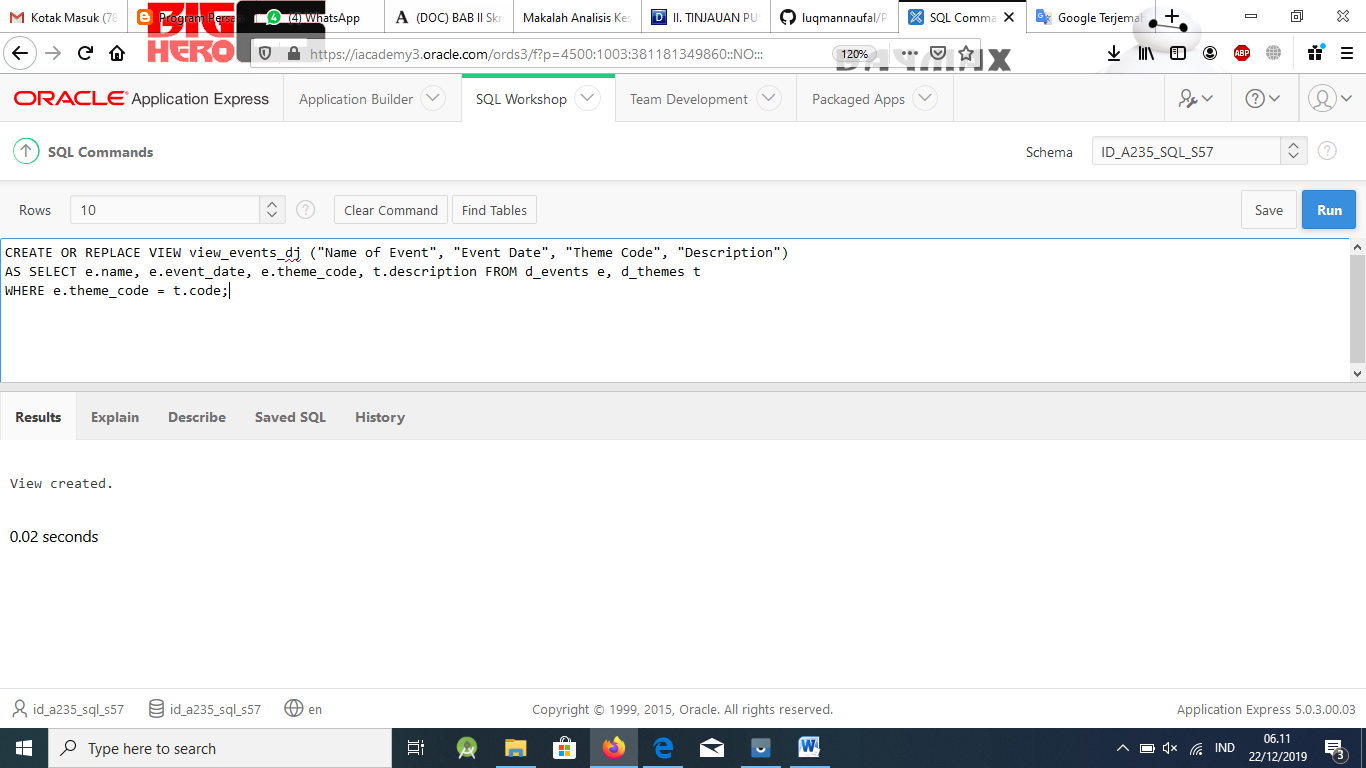
1. Jason Tsang, the disk jockey for DJs on Demand, needs a list of the past events and those planned for the coming months so he can make arrangements for each event’s equipment setup. As the company manager, you do not want him to have access to the price that clients paid for their events. Create a view for Jason to use that displays the name of the event, the event date, and the theme description. Use aliases for each col- umn name.

### Solution:

CREATE OR REPLACE VIEW view\_events\_dj ("Name of Event", "Event Date", "Theme Code", "Description")

AS SELECT e.name, e.event\_date, e.theme\_code, t.description FROM d\_events e, d\_themes t

WHERE e.theme\_code = t.code;



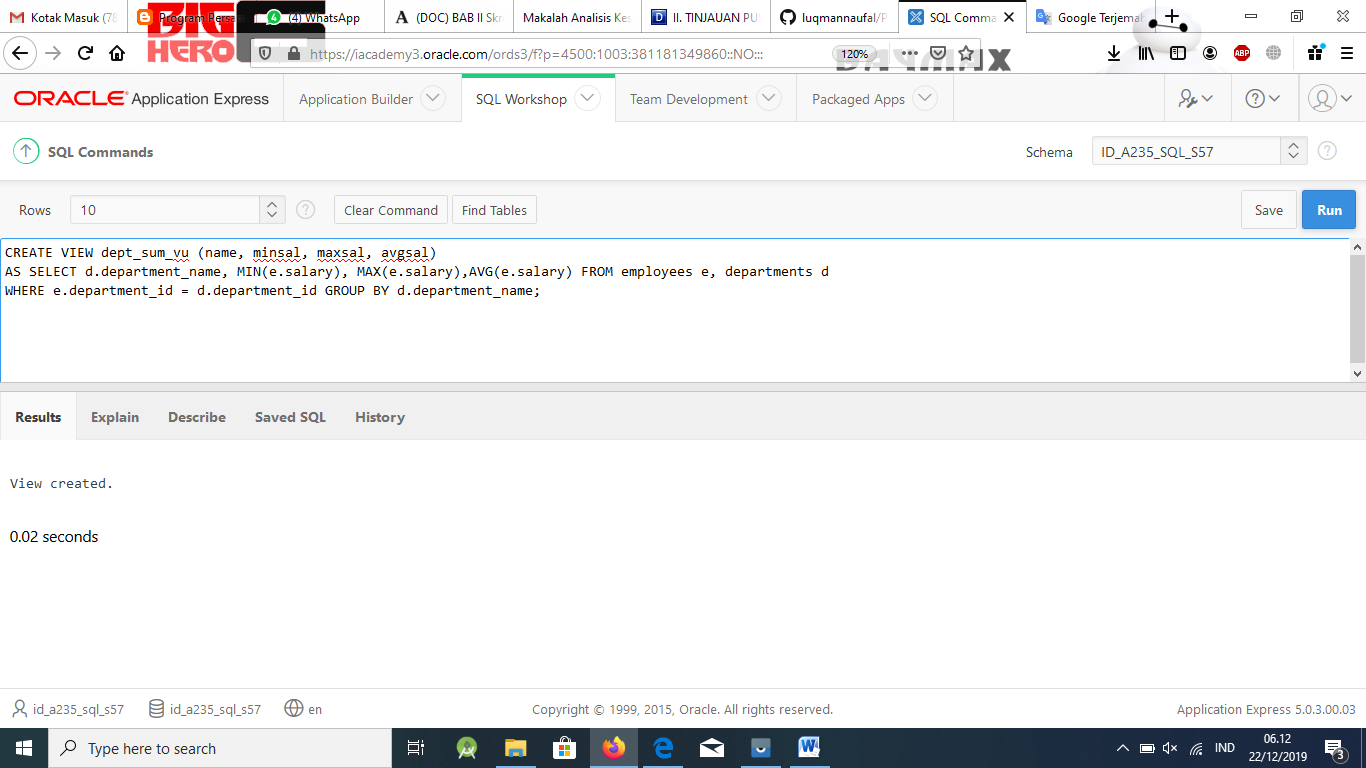
1. It is company policy that only upper-level management be allowed access to individual employee salaries. The department managers, however, need to know the minimum, max- imum, and average salaries, grouped by department. Use the Oracle database to prepare a view that displays the needed information for department managers.

### Solution:

CREATE VIEW dept\_sum\_vu (name, minsal, maxsal, avgsal)

AS SELECT d.department\_name, MIN(e.salary), MAX(e.salary),AVG(e.salary) FROM employees e, departments d

WHERE e.department\_id = d.department\_id GROUP BY d.department\_name;



[www.oracle.com/academy](http://www.oracle.com/academy)

Database Programming with SQL 15-2: DML Operations and Views Practice Solutions

# Vocabulary

Directions: Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **ROWNUM** | A pseudo-column which assigns a sequential value  starting with 1 to each of the rows returned from the subquery |
| **WITH CHECK OPTION** | Specifies that INSERTS and UPDATES performed through the view can’t create rows which the view cannot select |
| **WITH READ ONLY** | Ensures that no DML operations can be performed on  this view |

# Try It / Solve It

Use the DESCRIBE statement to verify that you have tables named copy\_d\_songs, copy\_d\_events, copy\_d\_cds, and copy\_d\_clients in your schema. If you don’t, write a query to create a copy of each.

1. Query the data dictionary USER\_UPDATABLE\_COLUMNS to make sure the columns in the base tables will allow UPDATE, INSERT, or DELETE. Use a SELECT statement or the Browse Data Dictionary feature in HTML DB. All table names in the data dictionary are stored in uppercase.

## Solution:

SELECT \*

FROM USER\_UPDATABLE\_COLUMNS

WHERE table\_name = 'COPY\_D\_SONGS';

Use the same syntax but change table\_name of the other tables.

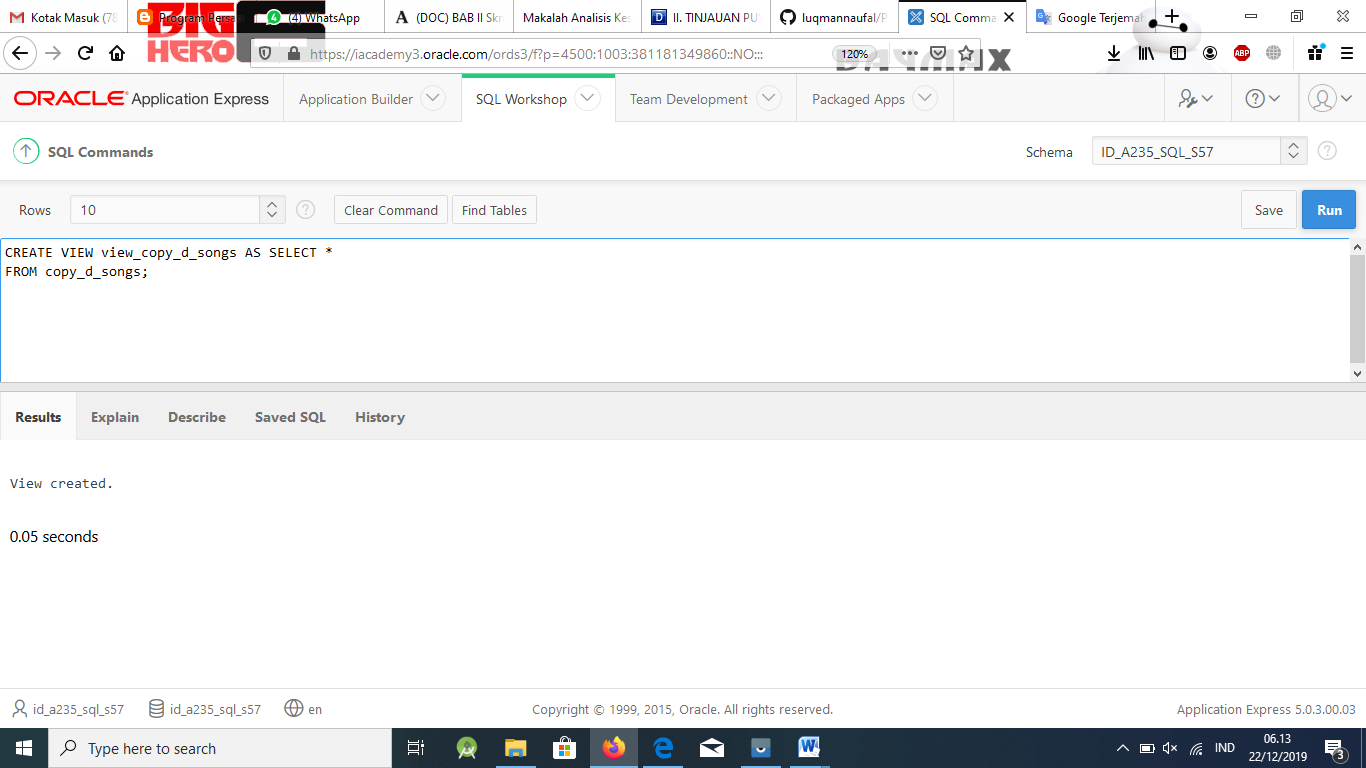


1. Use the CREATE or REPLACE option to create a view of *all* the columns in the copy\_d\_songs table called view\_copy\_d\_songs.

## Solution:

CREATE VIEW view\_copy\_d\_songs AS SELECT \*

FROM copy\_d\_songs;



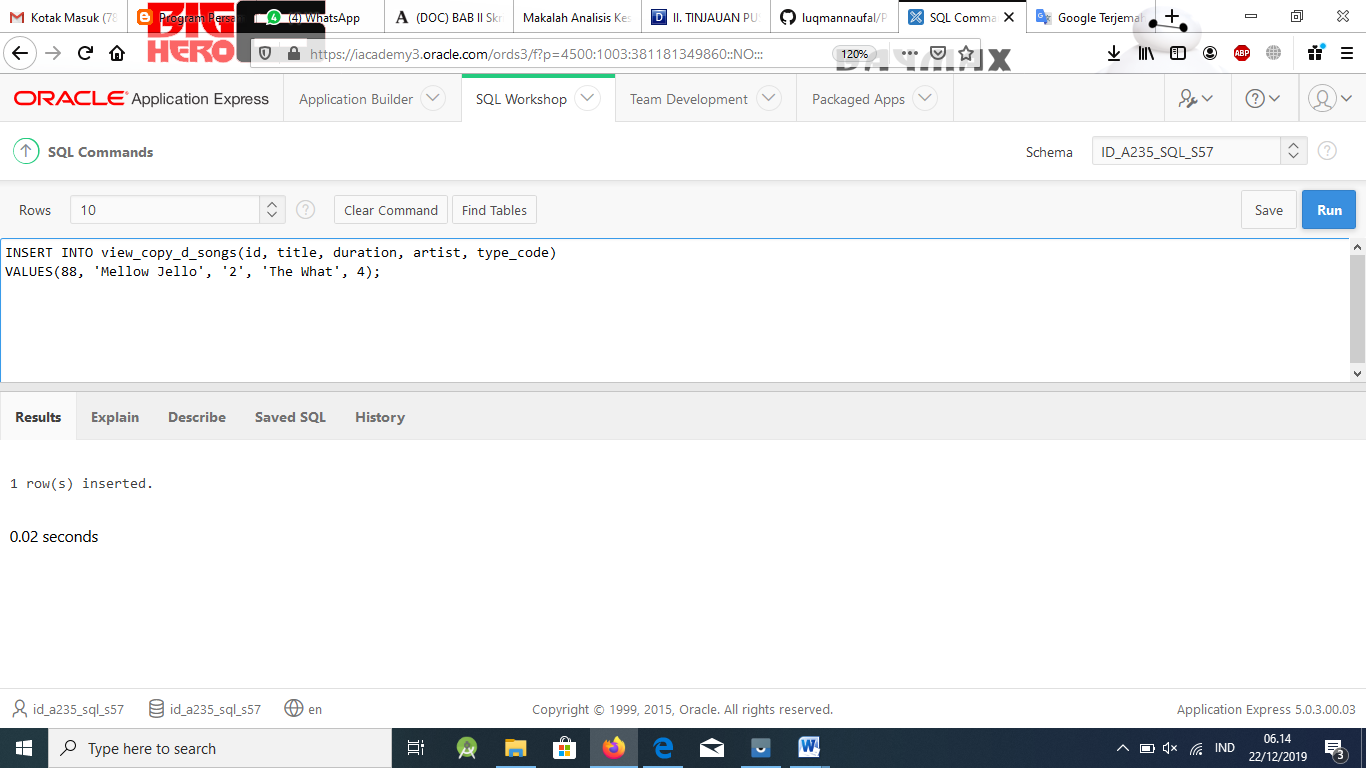
1. Use view\_copy\_d\_songs to INSERT the following data into the underlying copy\_d\_songs table. Execute a SELECT \* from copy\_d\_songs to verify your DML command. See the graphic.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | TITLE | DURATION | ARTIST | TYPE\_CODE |
| 88 | Mello Jello | 2 | The What | 4 |

## Solution:

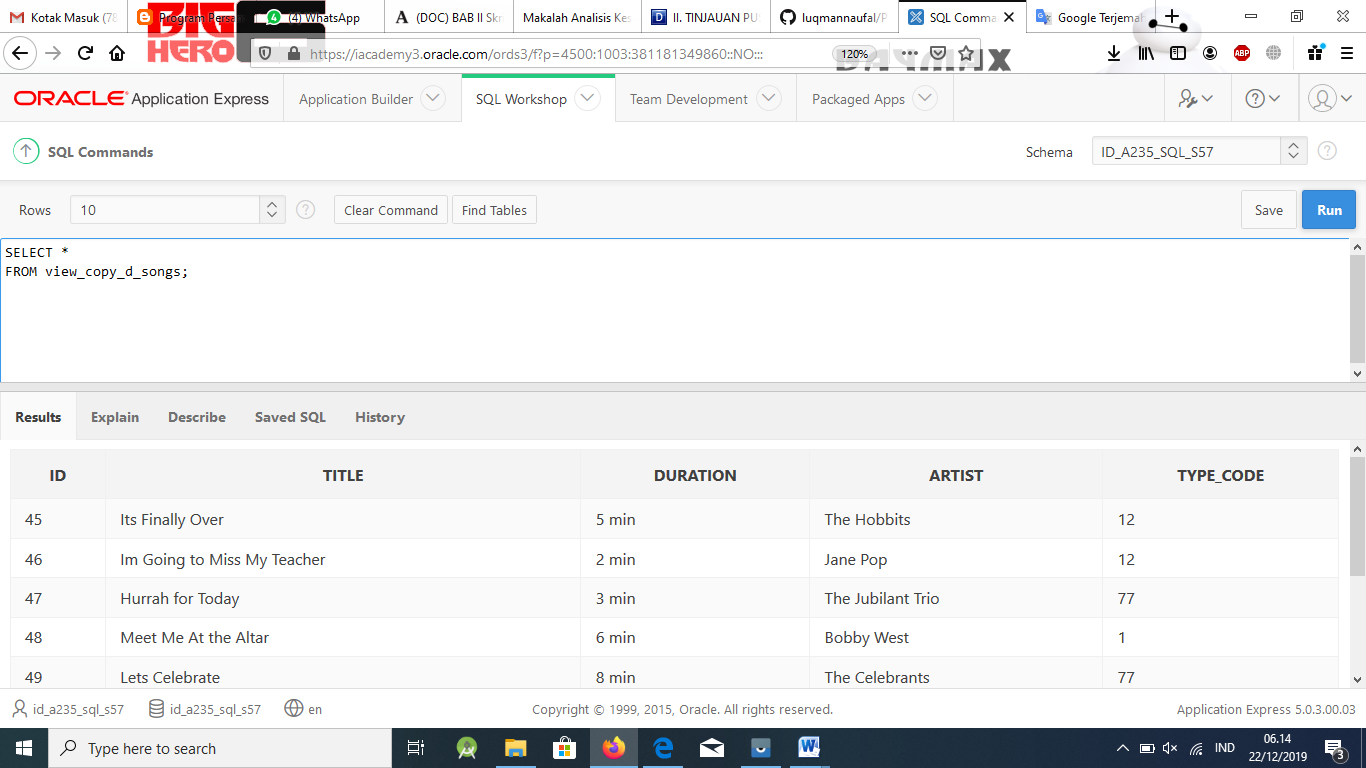
INSERT INTO view\_copy\_d\_songs(id, title, duration, artist, type\_code)

VALUES(88, 'Mellow Jello', '2', 'The What', 4);



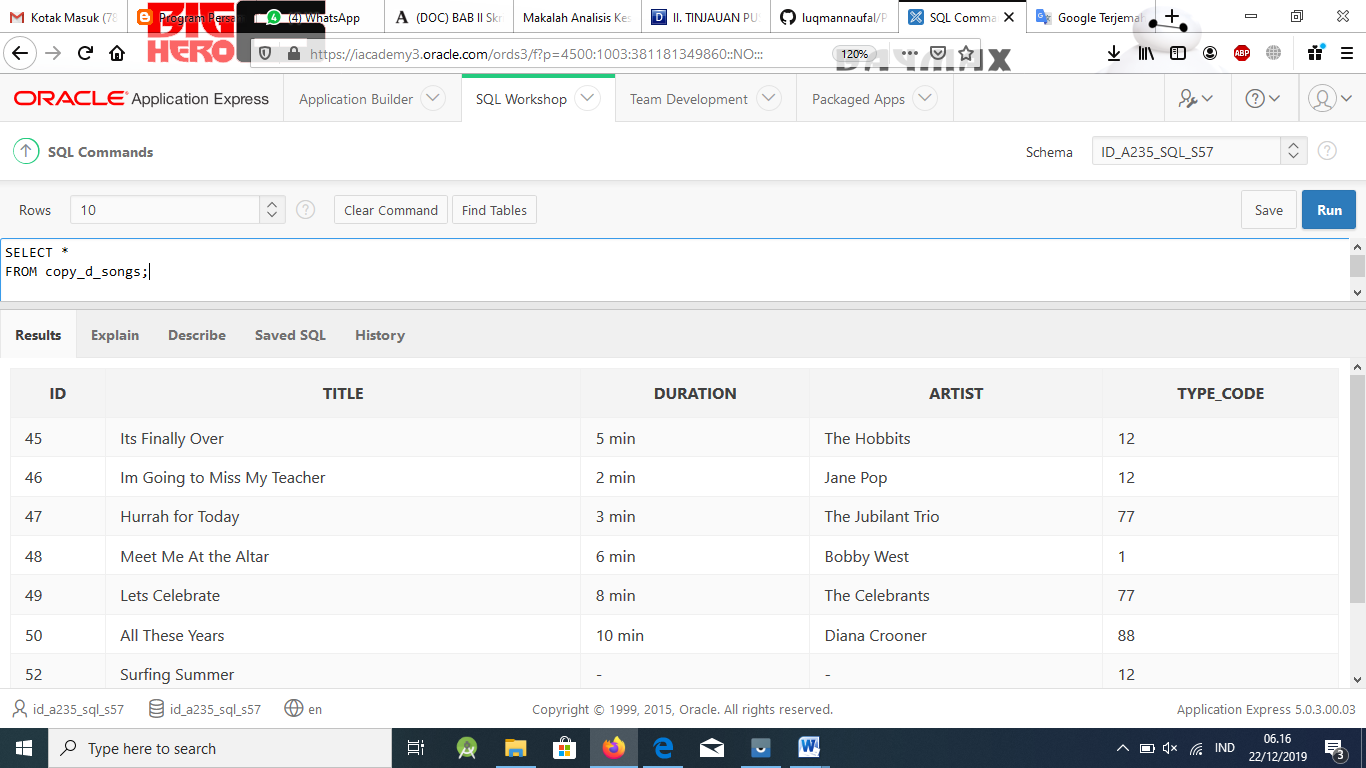
SELECT \*

FROM view\_copy\_d\_songs;



SELECT \*

FROM copy\_d\_songs;



1. Create a view based on the DJs on Demand COPY\_D\_CDS table. Name the view read\_copy\_d\_cds. Select all columns to be included in the view. Add a WHERE clause to restrict the year to 2000. Add the WITH READ ONLY option.

## Solution:

CREATE VIEW read\_copy\_d\_cds AS SELECT \*

FROM copy\_d\_cds WHERE year = 2000 WITH READ ONLY;



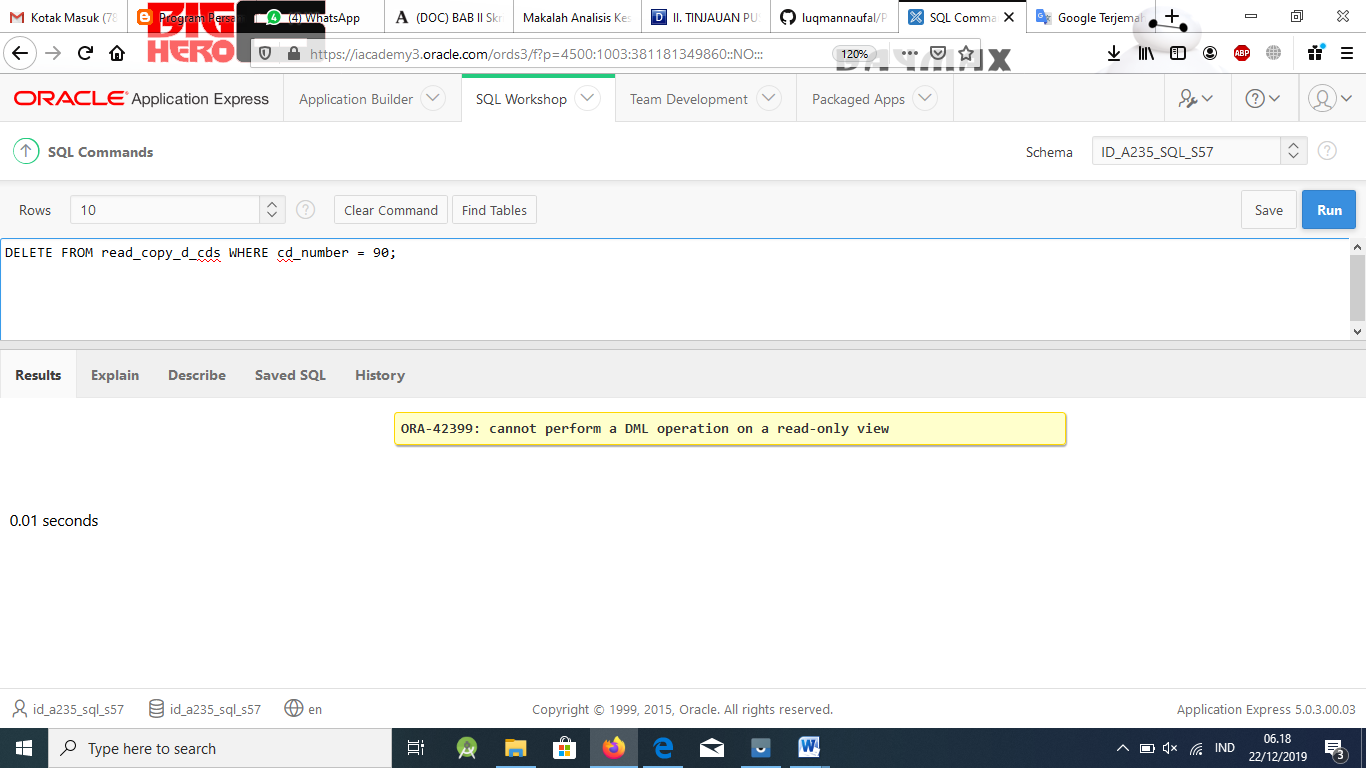
1. Using the read\_copy\_d\_cds view, execute a DELETE FROM read\_copy\_d\_cds WHERE cd\_number = 90;

## Solution:

DELETE FROM read\_copy\_d\_cds WHERE cd\_number = 90;

\*\*fail to execute as view is read only

ORA-01752: cannot delete from view without exactly one key-preserved table



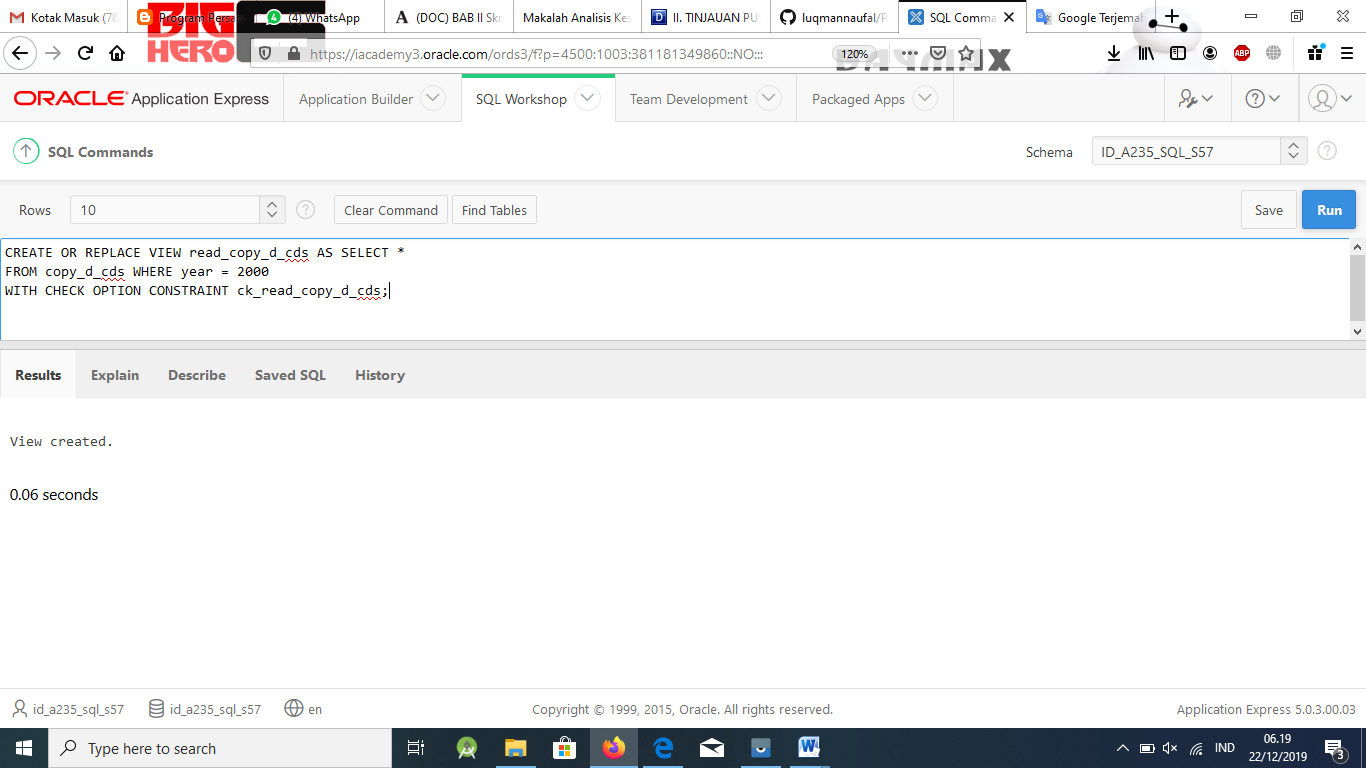
1. Use REPLACE to modify read\_copy\_d\_cds. Replace the READ ONLY option with WITH CHECK OPTION CONSTRAINT ck\_read\_copy\_d\_cds. Execute a SELECT \* statement to verify that the view exists.

## Solution:

CREATE OR REPLACE VIEW read\_copy\_d\_cds AS SELECT \*

FROM copy\_d\_cds WHERE year = 2000

WITH CHECK OPTION CONSTRAINT ck\_read\_copy\_d\_cds;

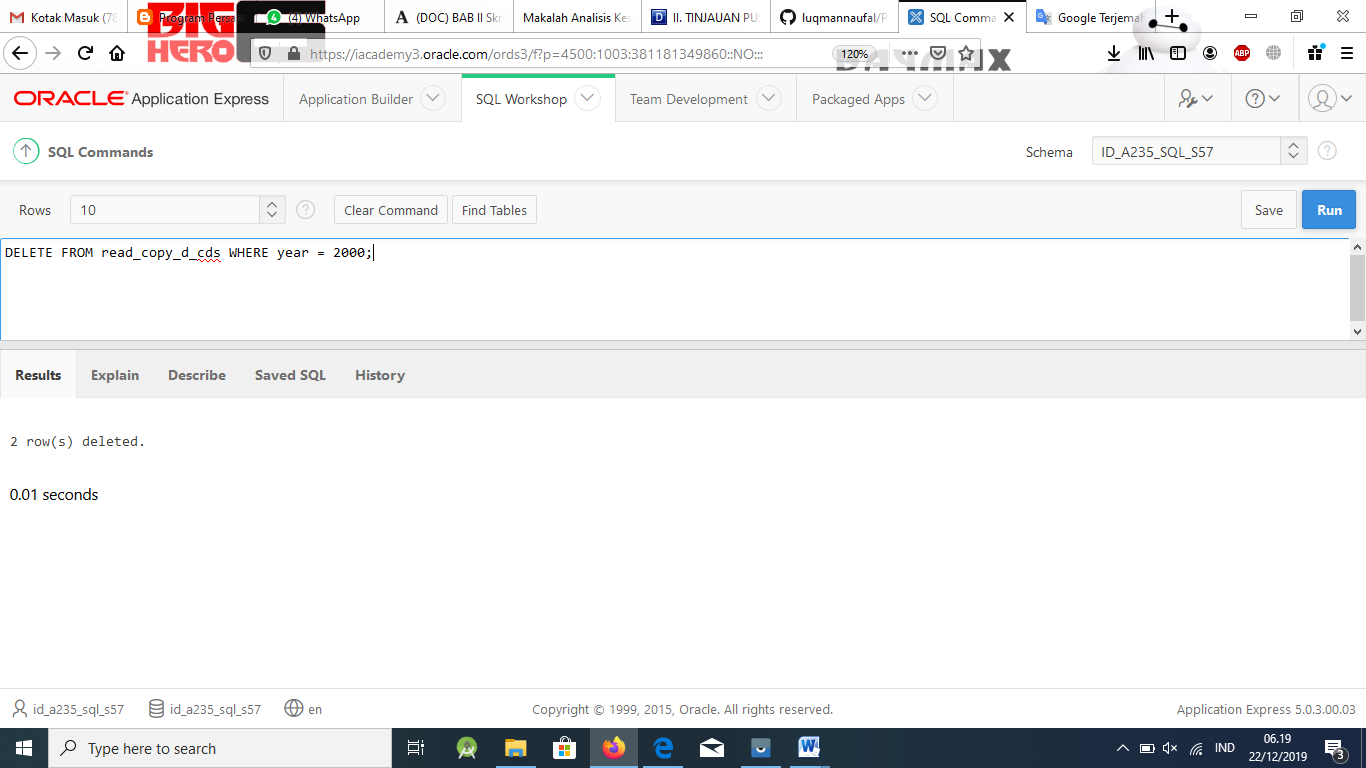


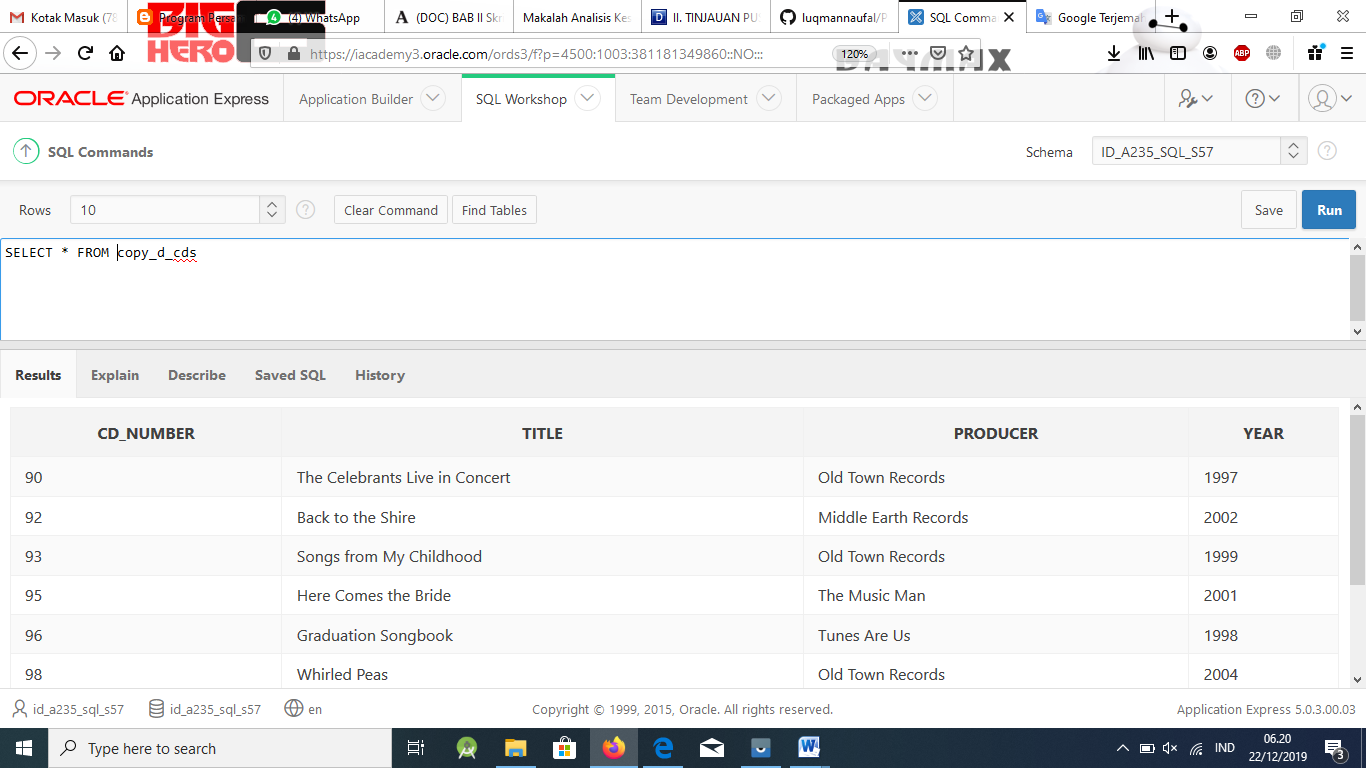
1. Use the read\_copy\_d\_cds view to delete any CD of year 2000 from the underlying copy\_d\_cds.

## Solution:

DELETE FROM read\_copy\_d\_cds WHERE year = 2000;

\*\*executes as it meets the CHECK OPTION criteria



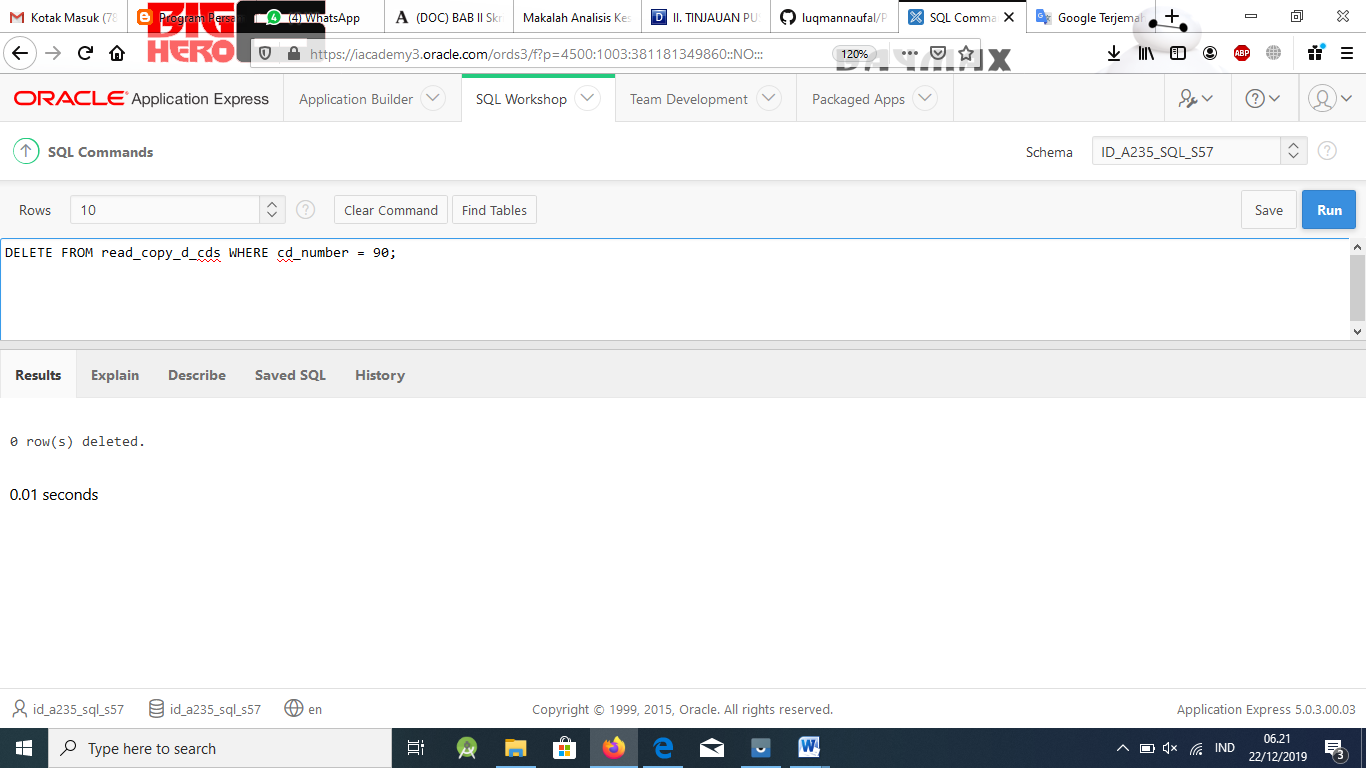


1. Use the read\_copy\_d\_cds view to delete cd\_number 90 from the underlying copy\_d\_cds table.

## Solution:

DELETE FROM read\_copy\_d\_cds WHERE cd\_number = 90;

\*\*query executes with 0 rows deleted

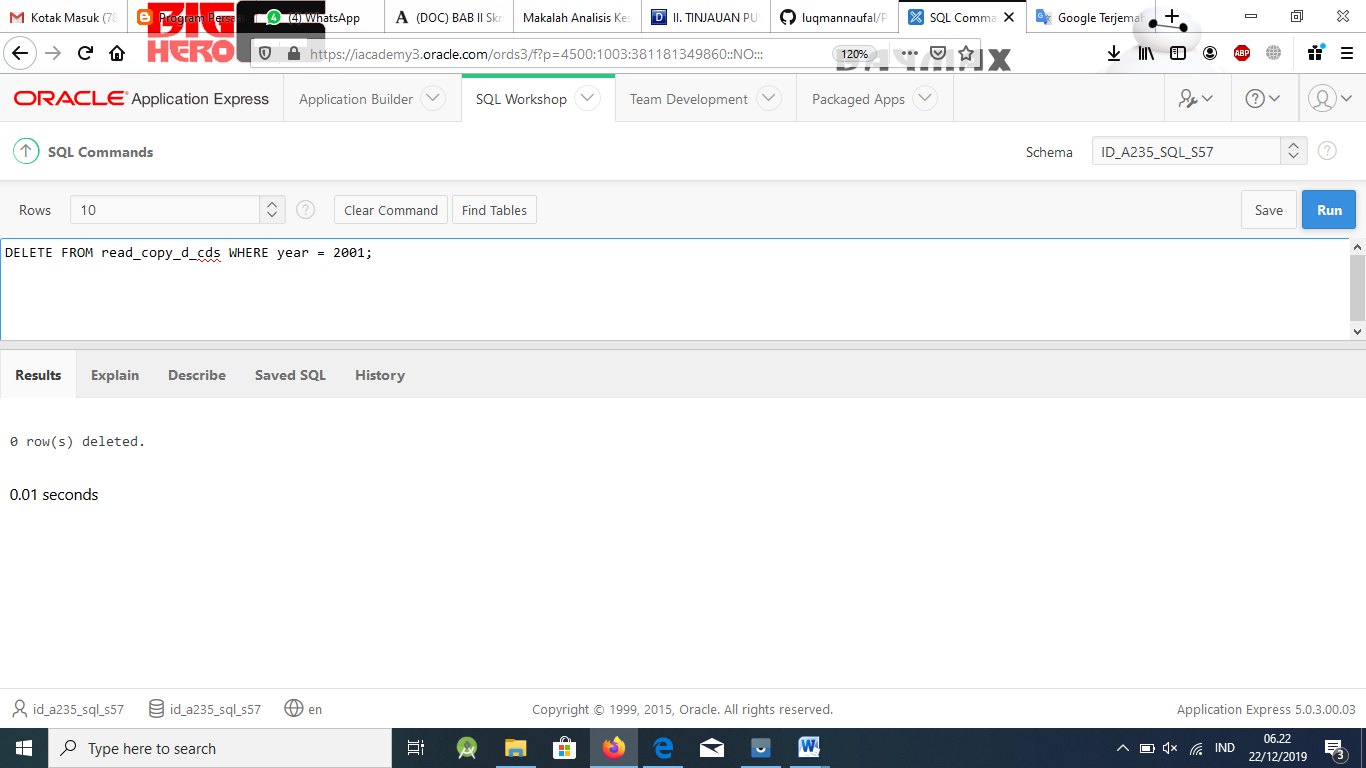


1. Use the read\_copy\_d\_cds view to delete year 2001 records.

## Solution:

DELETE FROM read\_copy\_d\_cds WHERE year = 2001;

\*\*query executes with 0 rows deleted



1. Execute a SELECT \* statement for the base table copy\_d\_cds. What rows were deleted?

## Solution:

The base tables should have all original rows with no year 2000 data.

1. What are the restrictions on modifying data through a view?

## Solution:

A row cannot be removed if the row contains:

* + Group functions
  + A GROUP BY clause
  + The DISTINCT keyword
  + The pseudo-column ROWNUM Keyword

Data cannot be modified if the view contains:

* + Group functions
  + A GROUP BY clause
  + The DISTINCT keyword
  + The pseudo column ROWNUM keyword
  + Columns defined by expressions

Data cannot be added if the view:

* + Includes group functions
  + Includes a GROUP BY clause
  + Includes the DISTINCT keyword
  + Includes the pseudo column ROWNUM keyword
  + Includes columns defined by expressions
  + Does not include NOT NULL columns in the base tables

1. What is Moore’s Law? Do you consider that it will continue to apply indefinitely? Support your opinion with research from the internet.

## Solution:

Moore’s Law states that the number of transistors on a given chip can be doubled every two years.

1. What is the “singularity” in terms of computing?

## Solution:

The singularity – the argument that computing power will improve to the point that it has such an impact on human evolution that it is impossible to determine what will come after.

;

[www.oracle.com/academy](http://www.oracle.com/academy)

# Database Programming with SQL 15-3: Managing Views

Practice Solutions

## Vocabulary

Directions: Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **TOP- N-ANALYSIS** | Asks for the N largest or smallest values in a column |
| **DROP VIEW** | Removes a view |
| **INLINE VIEW** | Subquery with an alias that can be used within a SQL statement |

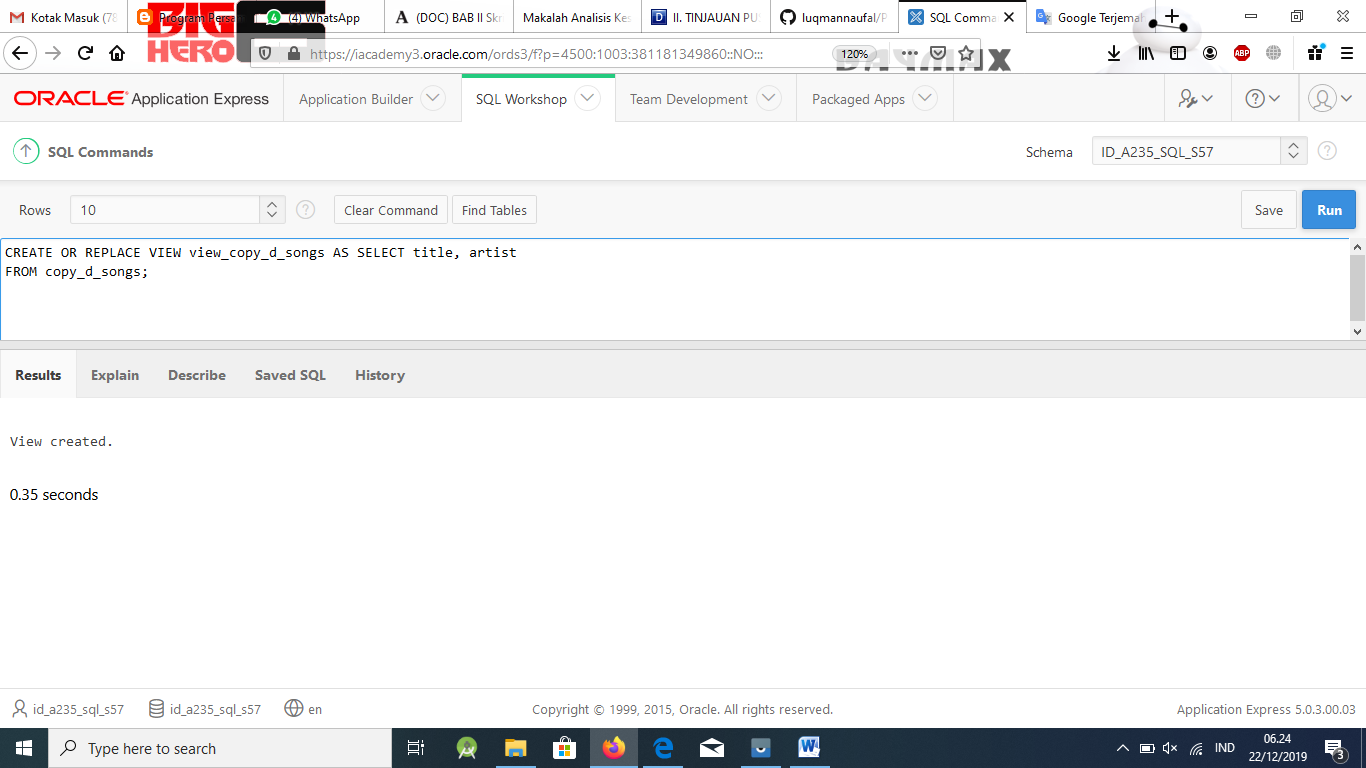
## Try It / Solve It

1. Create a view from the copy\_d\_songs table called view\_copy\_d\_songs that includes only the title and artist. Execute a SELECT \* statement to verify that the view exists.

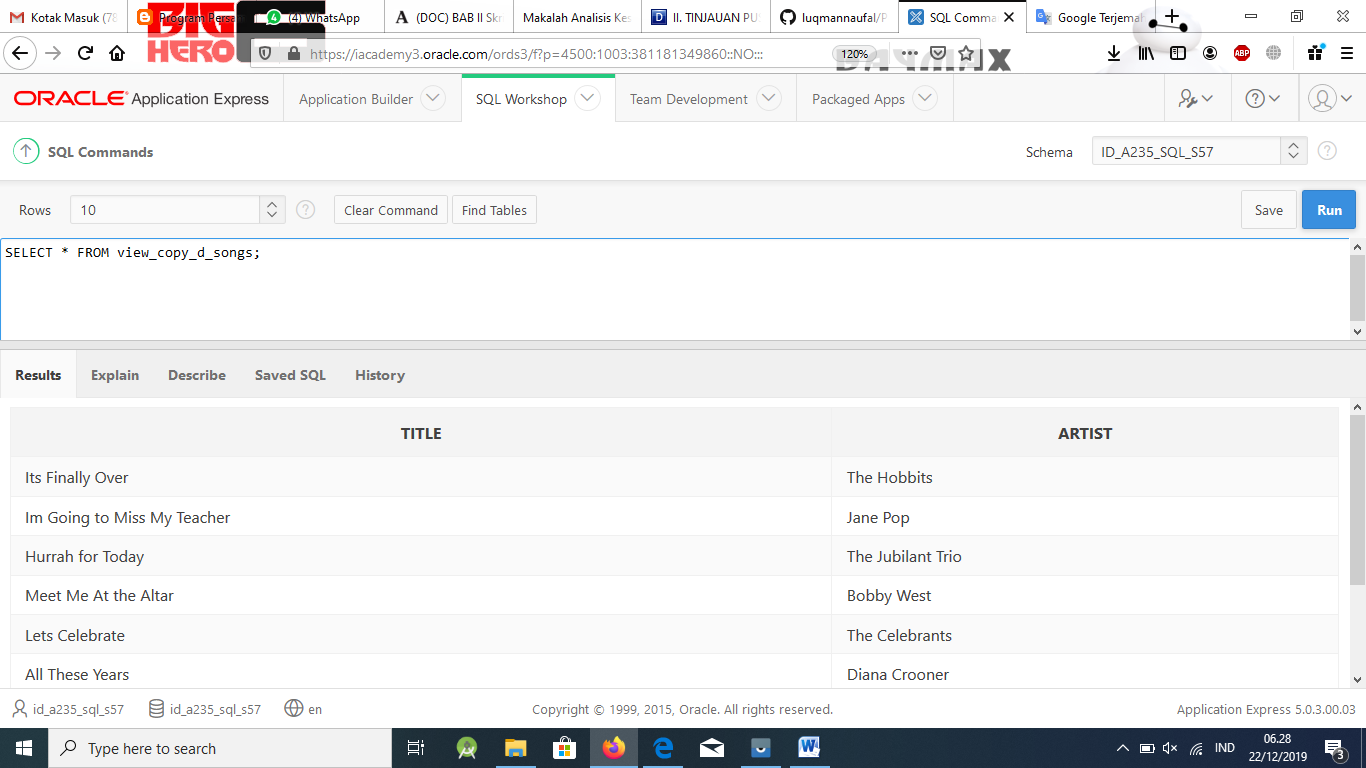
### Solution:

CREATE OR REPLACE VIEW view\_copy\_d\_songs AS SELECT title, artist

FROM copy\_d\_songs;



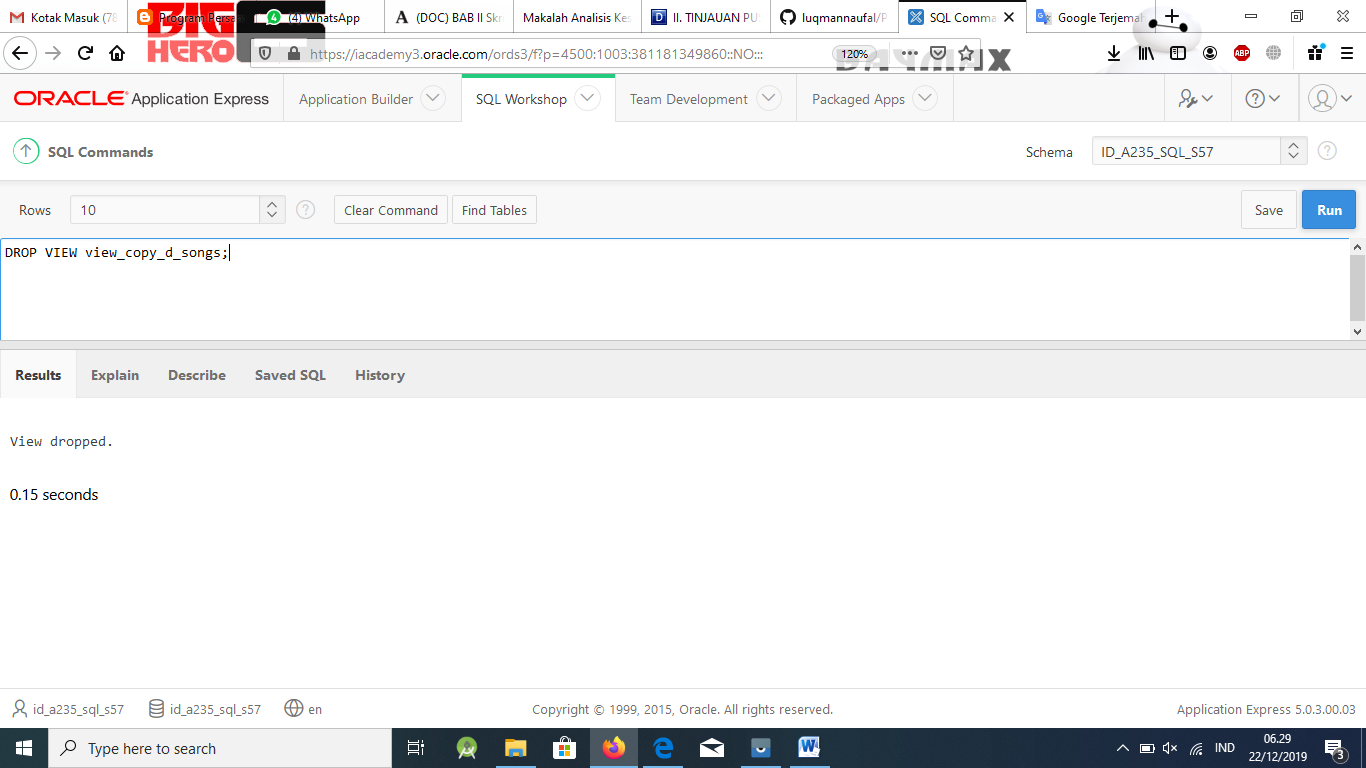
SELECT \* FROM view\_copy\_d\_songs;



1. Issue a DROP view\_copy\_d\_songs. Execute a SELECT \* statement to verify that the view has been deleted.

### Solution:

DROP VIEW view\_copy\_d\_songs;



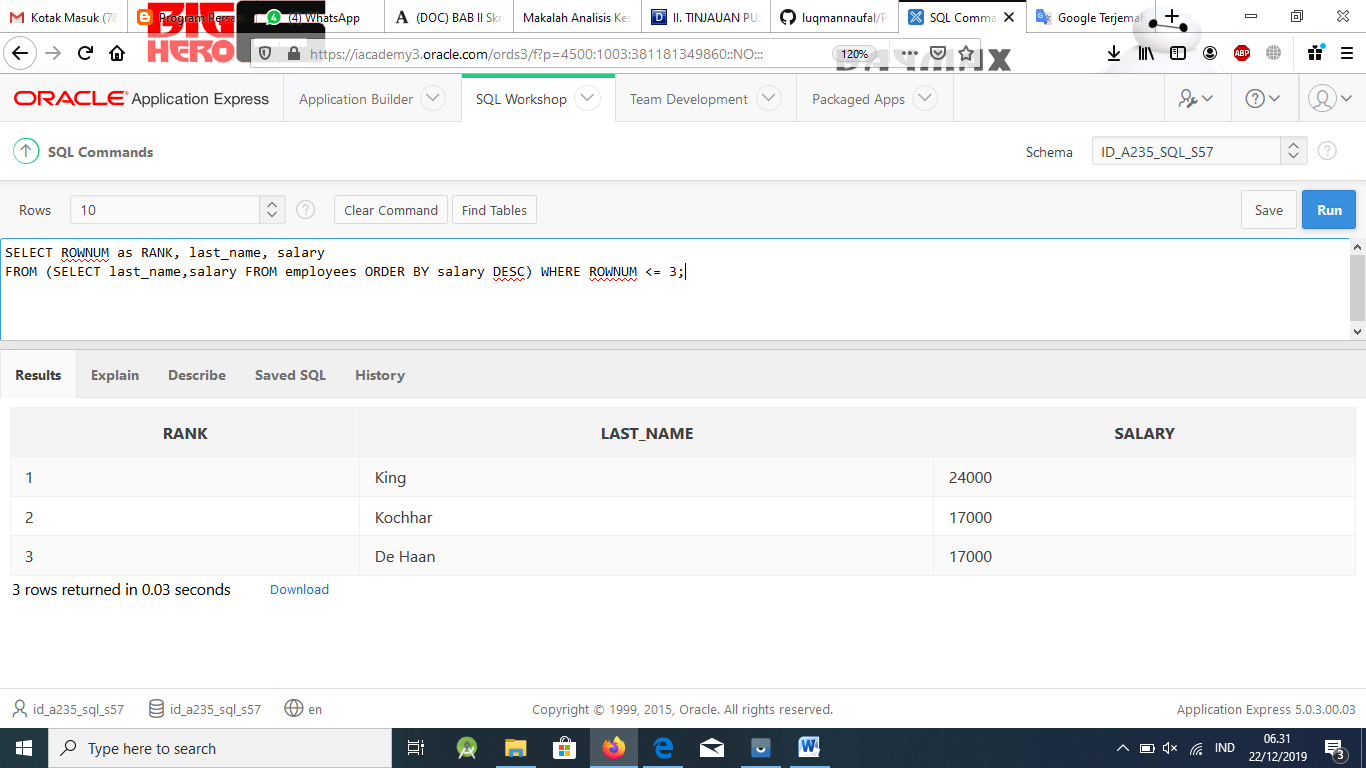
SELECT \* FROM view\_copy\_d\_songs;

1. Create a query that selects the last name and salary from the Oracle database. Rank the salaries from highest to lowest for the top three employees.

### Solution:

SELECT ROWNUM as RANK, last\_name, salary

FROM (SELECT last\_name,salary FROM employees ORDER BY salary DESC) WHERE ROWNUM <= 3;



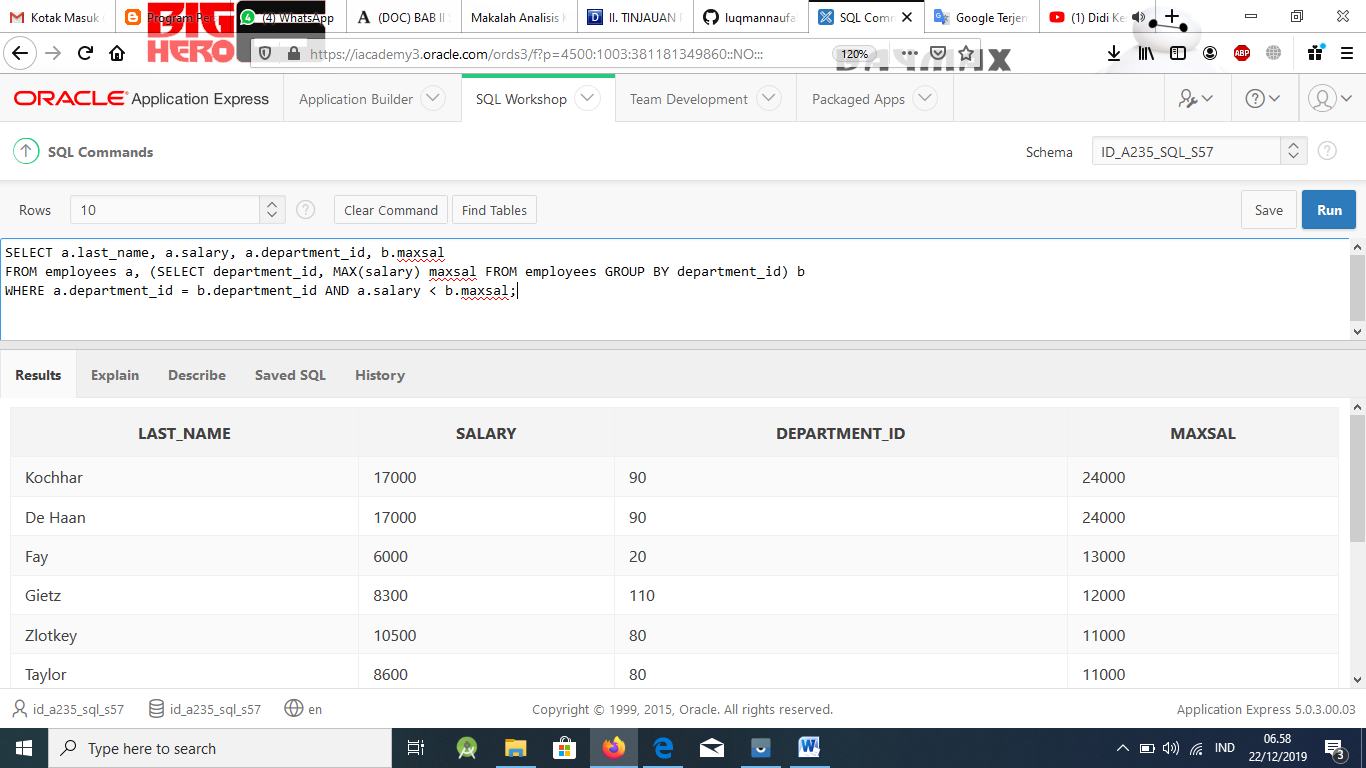
1. Construct an inline view from the Oracle database that lists the last name, salary, depart- ment ID, and maximum salary for each department. Hint: One query will need to calculate maximum salary by department ID.

### Solution:

SELECT a.last\_name, a.salary, a.department\_id, b.maxsal

FROM employees a, (SELECT department\_id, MAX(salary) maxsal FROM employees GROUP BY department\_id) b

WHERE a.department\_id = b.department\_id AND a.salary < b.maxsal;

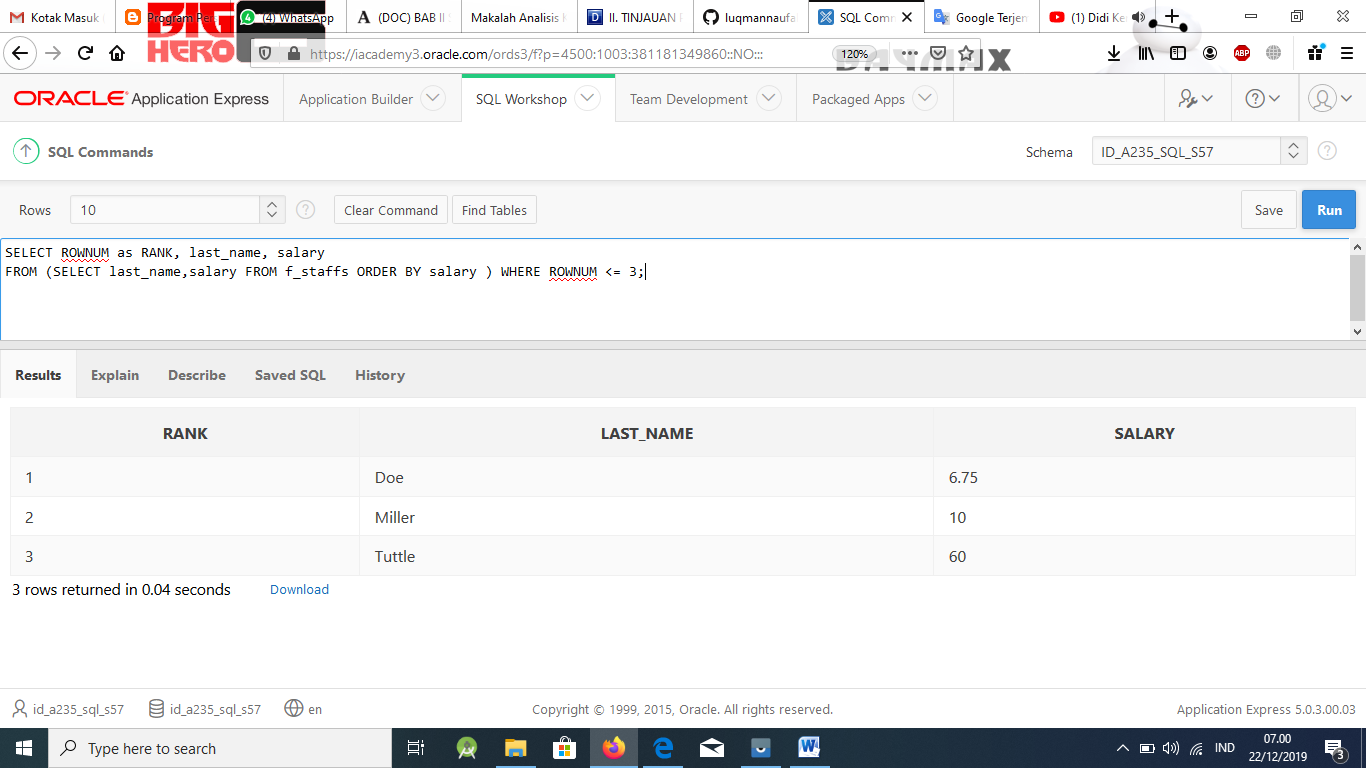


1. Create a query that will return the staff members of Global Fast Foods ranked by salary from lowest to highest.

### Solution:

SELECT ROWNUM as RANK, last\_name, salary

FROM (SELECT last\_name,salary FROM f\_staffs ORDER BY salary ) WHERE ROWNUM <= 3;



*Extension Exercises*

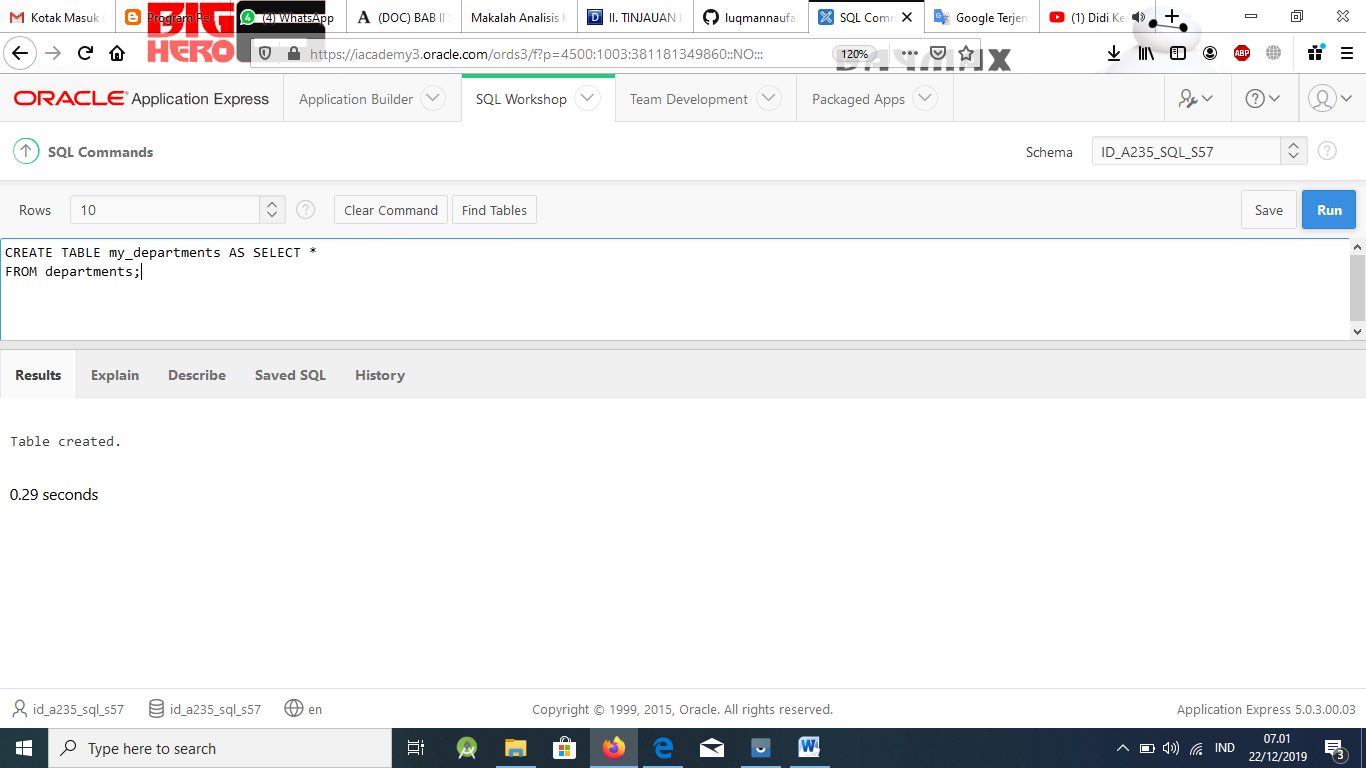
### NOTE: Students should complete each practice exercise in the order presented. Each subsequent exercise is an extension of the prior exercise. If students make a mistake, a view can be re-created easily with a CREATE or REPLACE VIEW state- ment. If students have difficulty, do one question at a time and review the process for setting up the query and the result set. Review with students the Study Guide and Vocabulary.

1. Create a new table called my\_departments and add all columns and all rows to it using a subquery from the Oracle departments table. Do a SELECT \* from my\_departments to confirm that you have all the columns and rows.

### Solution:

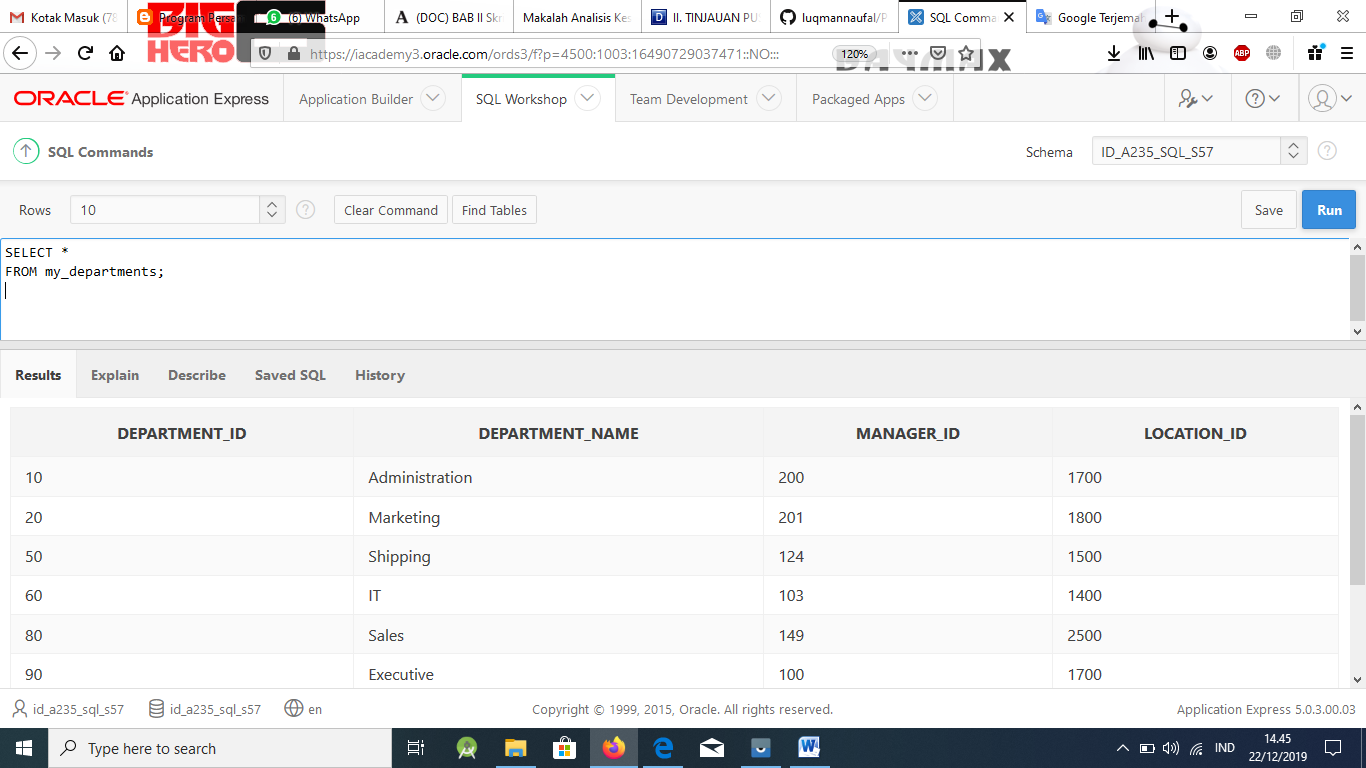
CREATE TABLE my\_departments AS SELECT \*

FROM departments;



SELECT \*

FROM my\_departments;

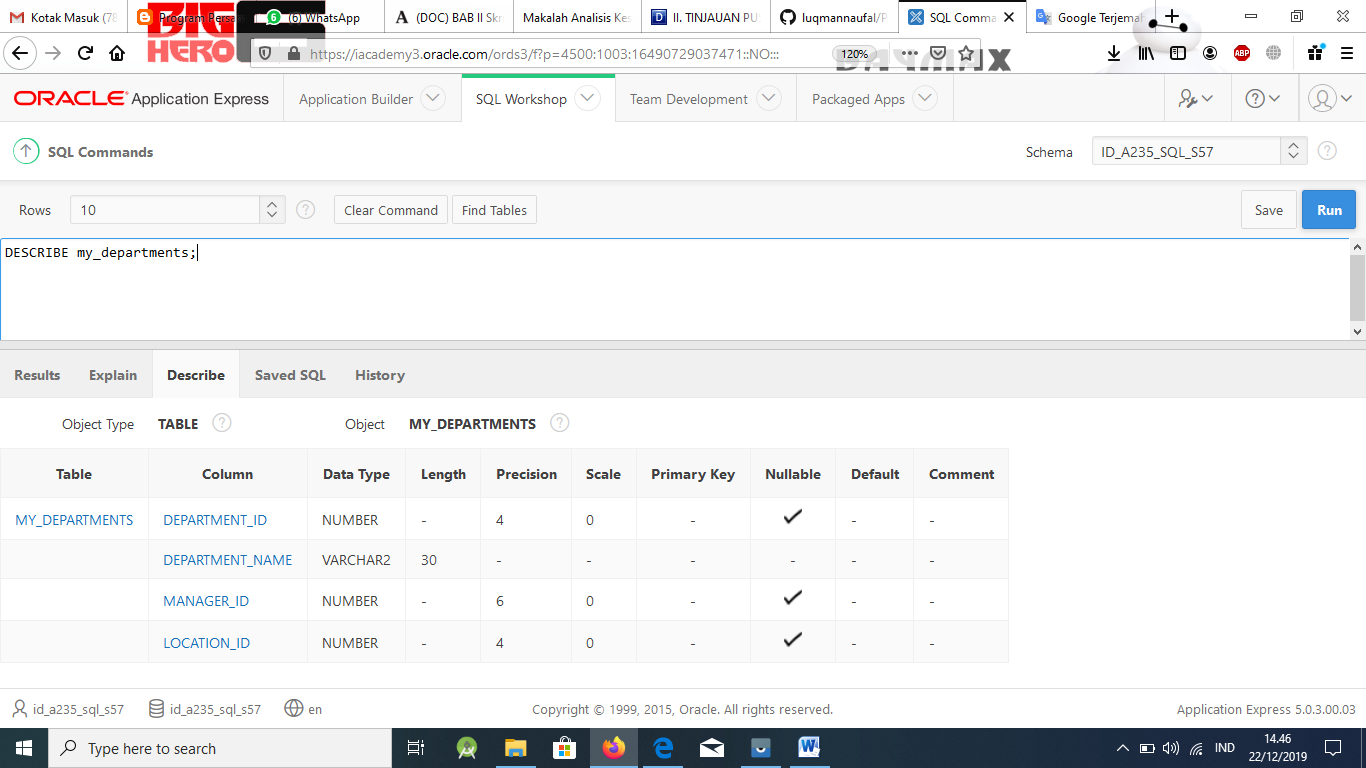


1. To view any constraints that may affect the my\_departments table, DESCRIBE my\_departments to check if any constraints were carried over from the departments table. If there are constraints on my\_departments, use an ALTER TABLE command to DISABLE all constraints on my\_departments.

### Solution:

DESCRIBE my\_departments;

\*\*department\_name is NOT NULL, constraint name is SYS\_C001868830 (this name could vary)



ALTER TABLE my\_departments

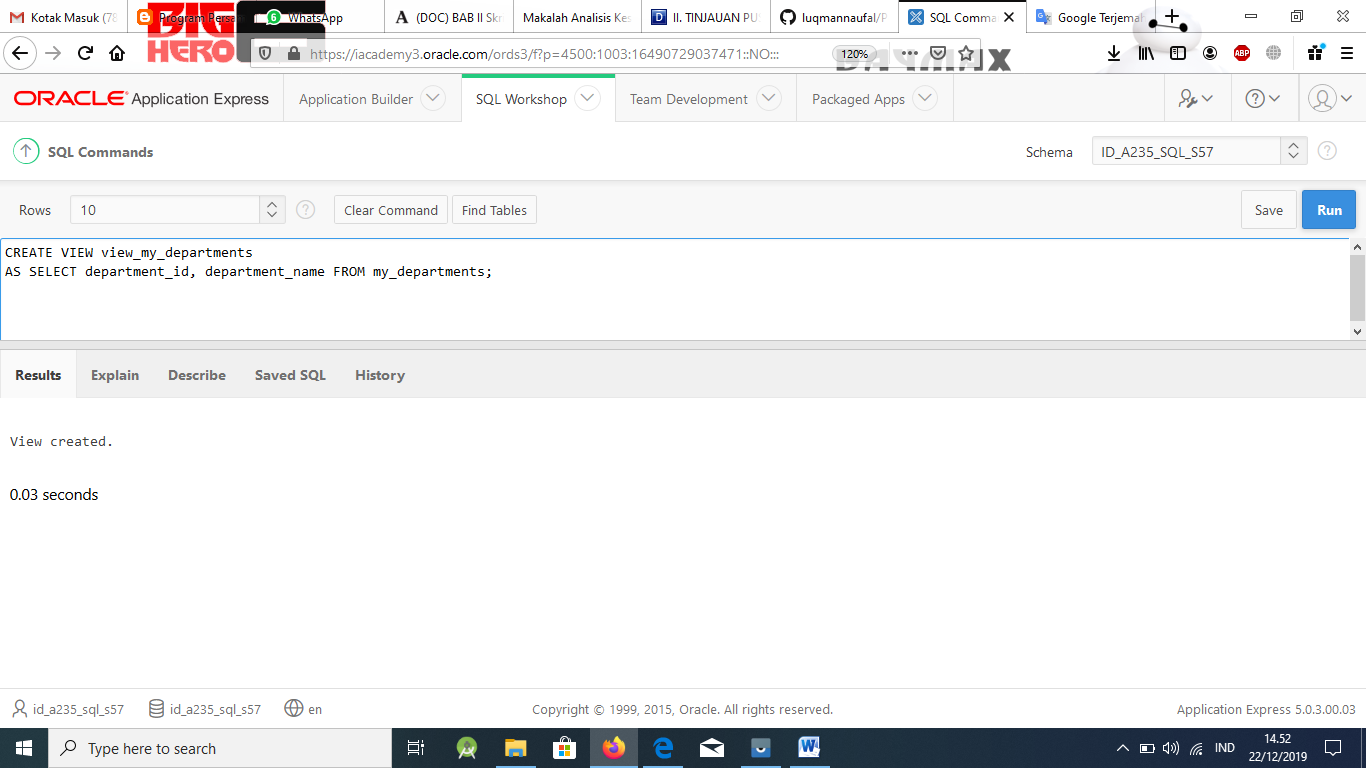
DISABLE CONSTRAINT SYS\_C001868830;

1. Create a view called view\_my\_departments that includes: department\_id and depart- ment\_name.

### Solution:

CREATE VIEW view\_my\_departments

AS SELECT department\_id, department\_name FROM my\_departments;

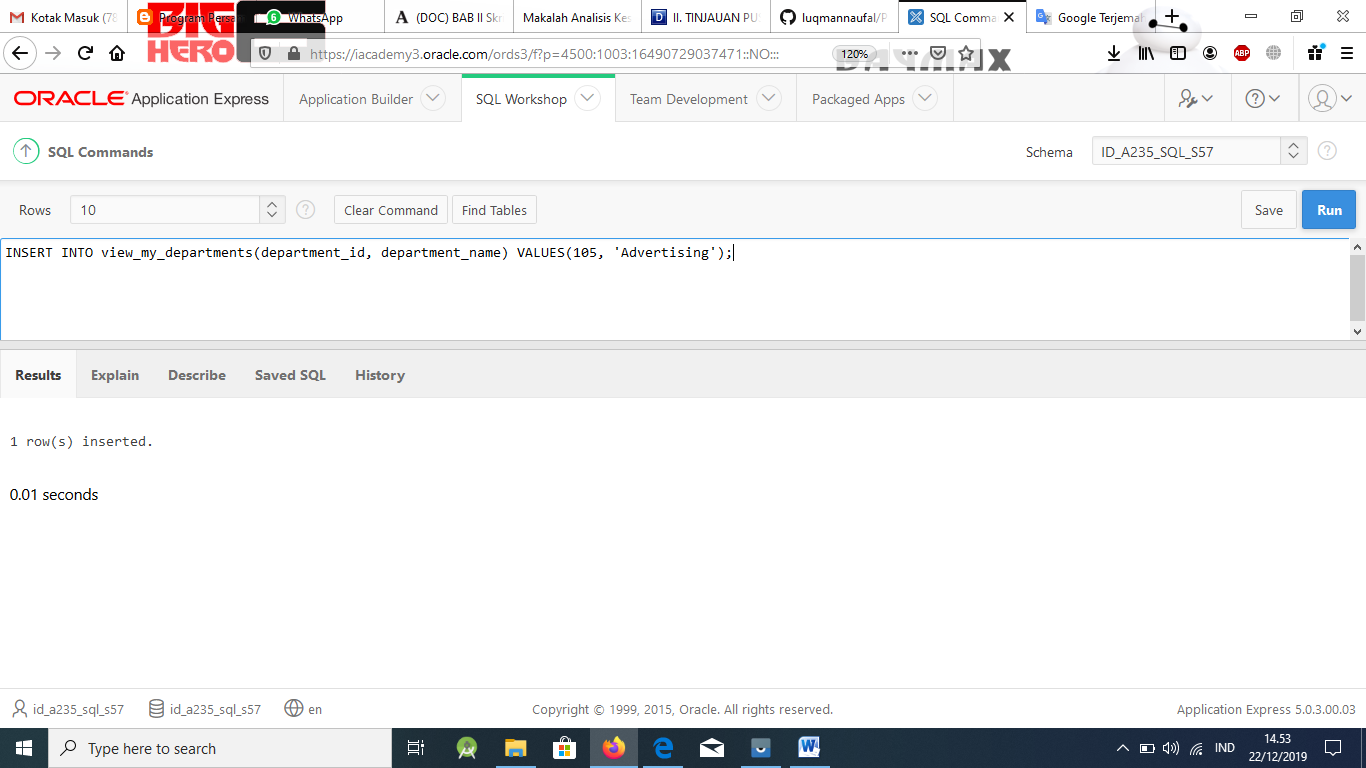


1. Add the following data to the my\_departments table using view\_my\_departments.

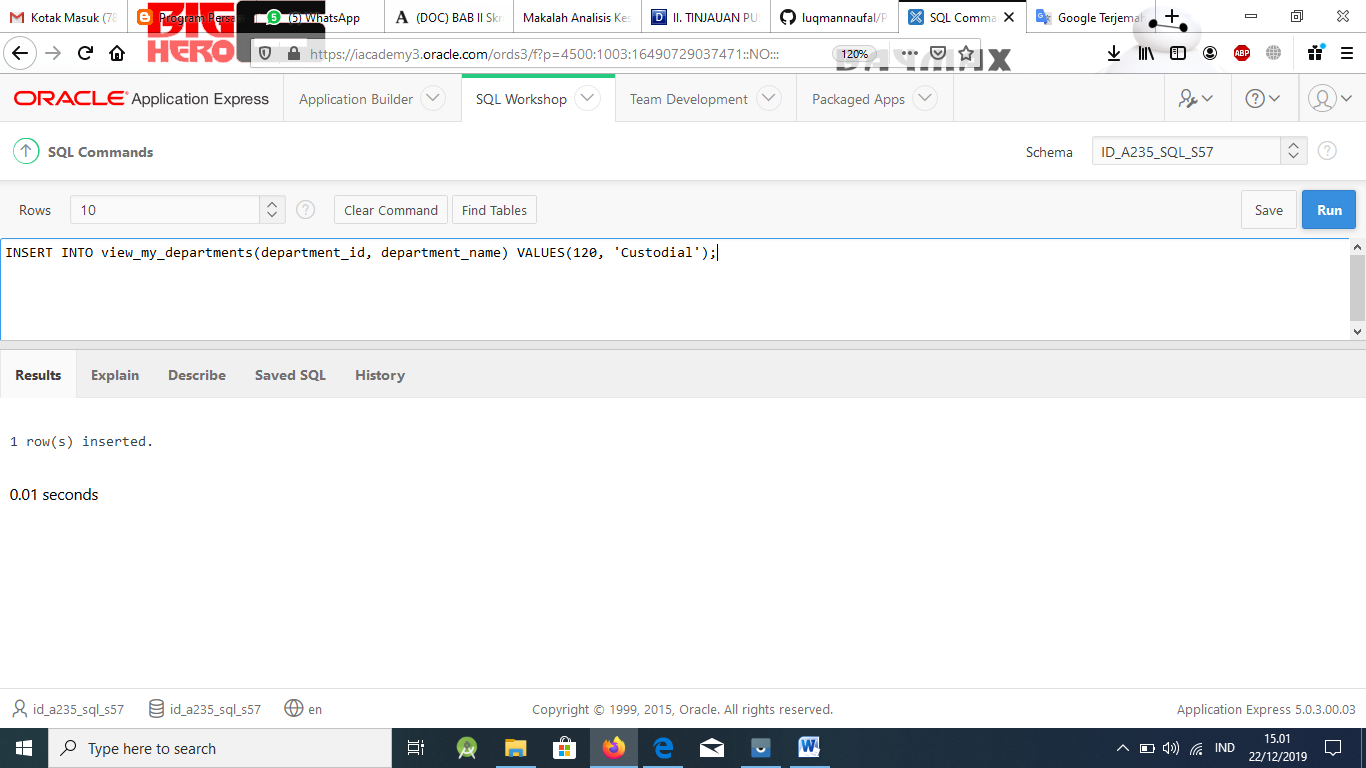
|  |  |
| --- | --- |
| department\_id | department\_name |
| 105 | Advertising |
| 120 | Custodial |
| 130 | Planning |

### Solution:

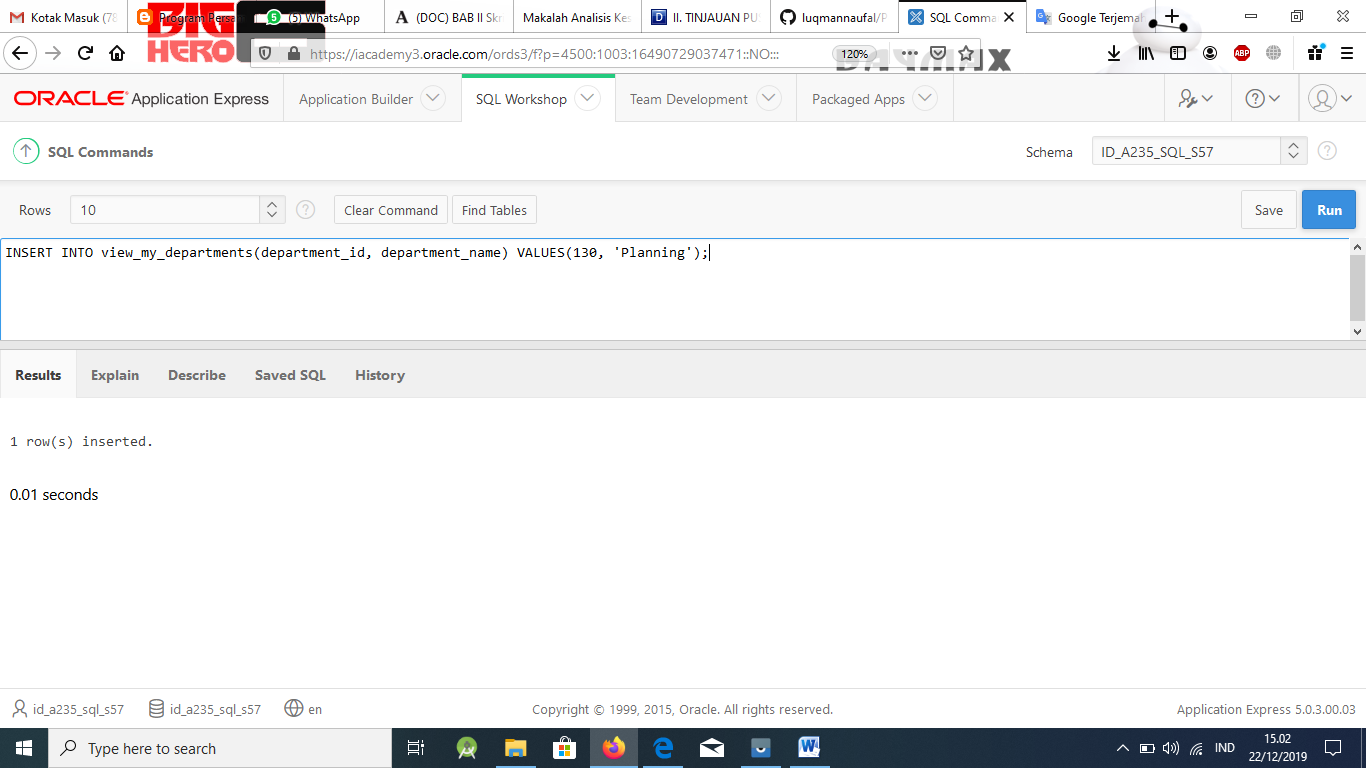
INSERT INTO view\_my\_departments(department\_id, department\_name) VALUES(105, 'Advertising');



INSERT INTO view\_my\_departments(department\_id, department\_name) VALUES(120, 'Custodial');



INSERT INTO view\_my\_departments(department\_id, department\_name) VALUES(130, 'Planning');

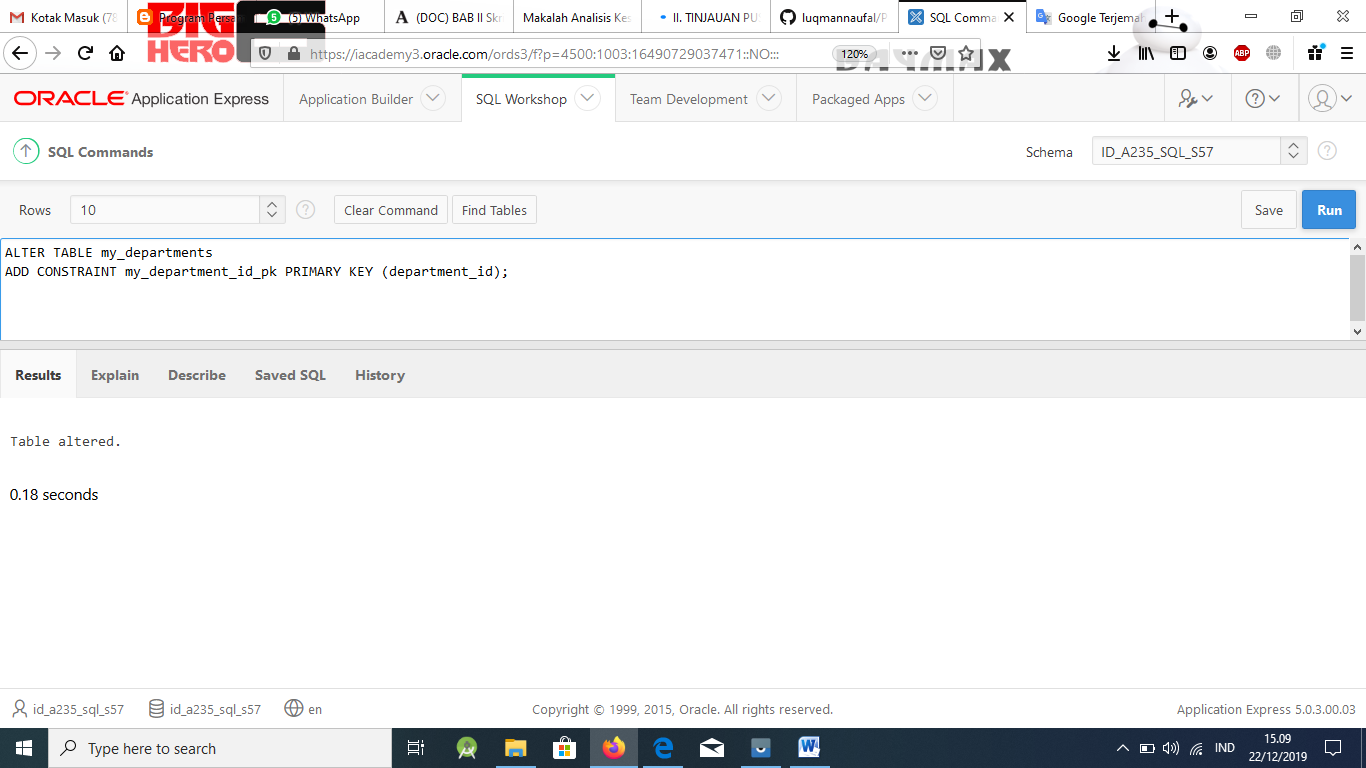


1. Create or enable the department\_id column as the primary key.

### Solution:

ALTER TABLE my\_departments

ADD CONSTRAINT my\_department\_id\_pk PRIMARY KEY (department\_id);

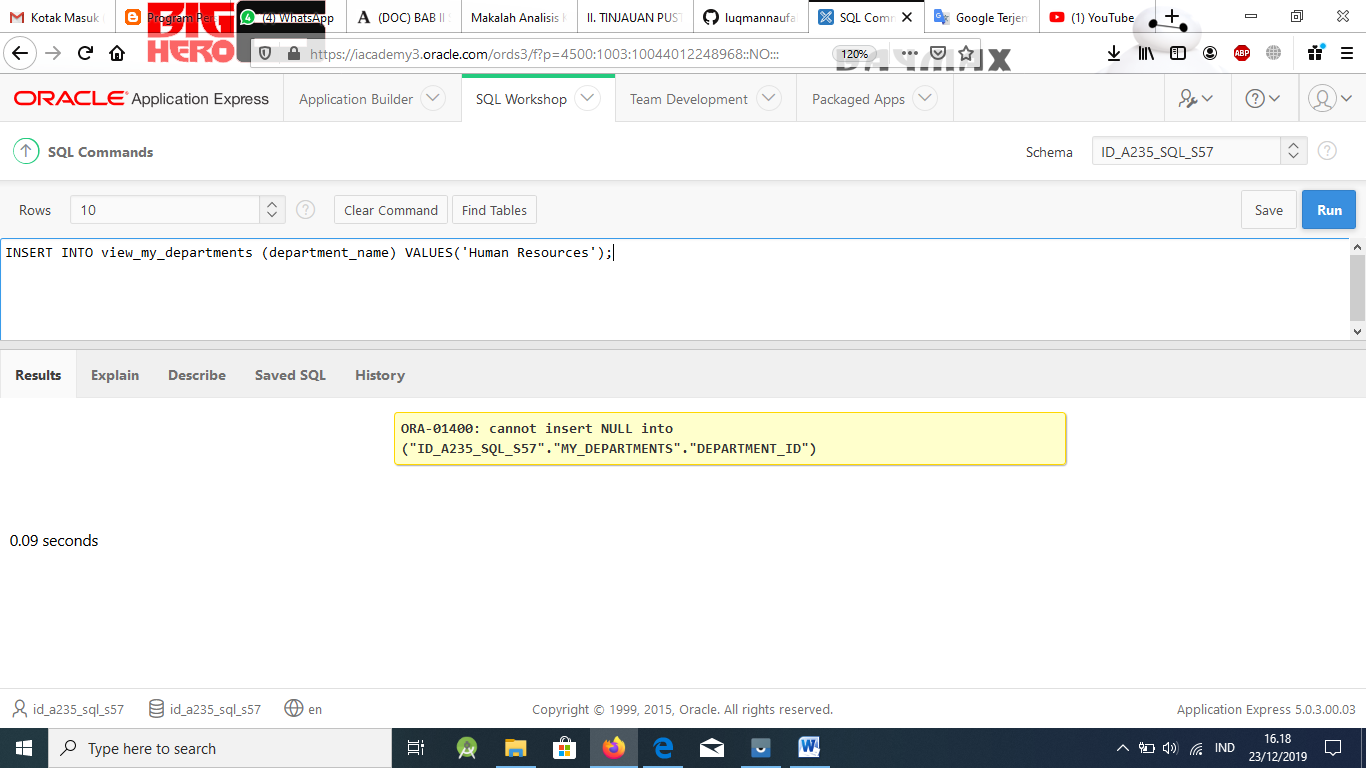


1. Enter a new department named Human Resources into the my\_departments table using view\_my\_departments. Do not add a new department ID.

### Solution:

INSERT INTO view\_my\_departments (department\_name) VALUES('Human Resources');

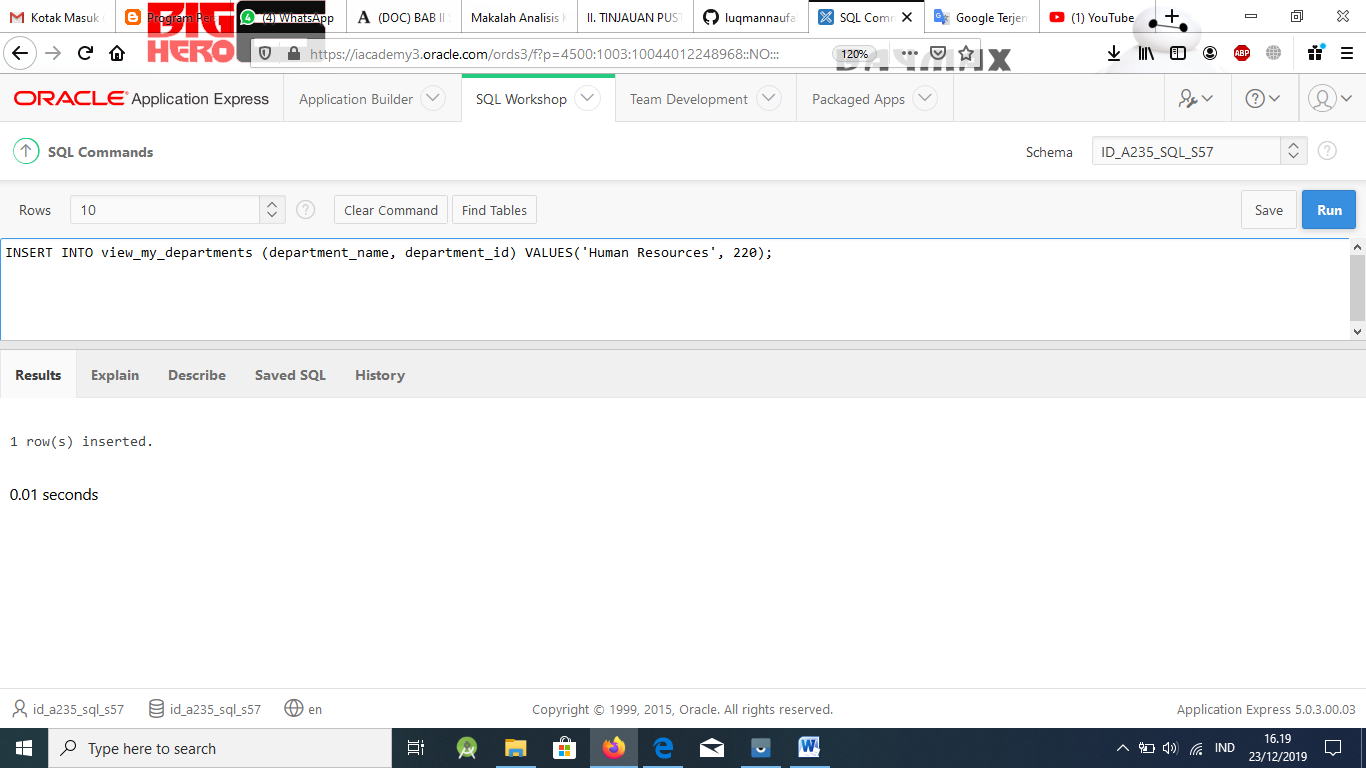
ORA-01400: cannot insert NULL into ("USWA\_SKHS\_SQL01\_T01"."DEPARTMENTS"."DEPARTMENT\_ID")



1. Add the Human Resources department, department ID 220, to my\_departments using view\_my\_departments.

### Solution:

INSERT INTO view\_my\_departments (department\_name, department\_id) VALUES('Human Resources', 220);

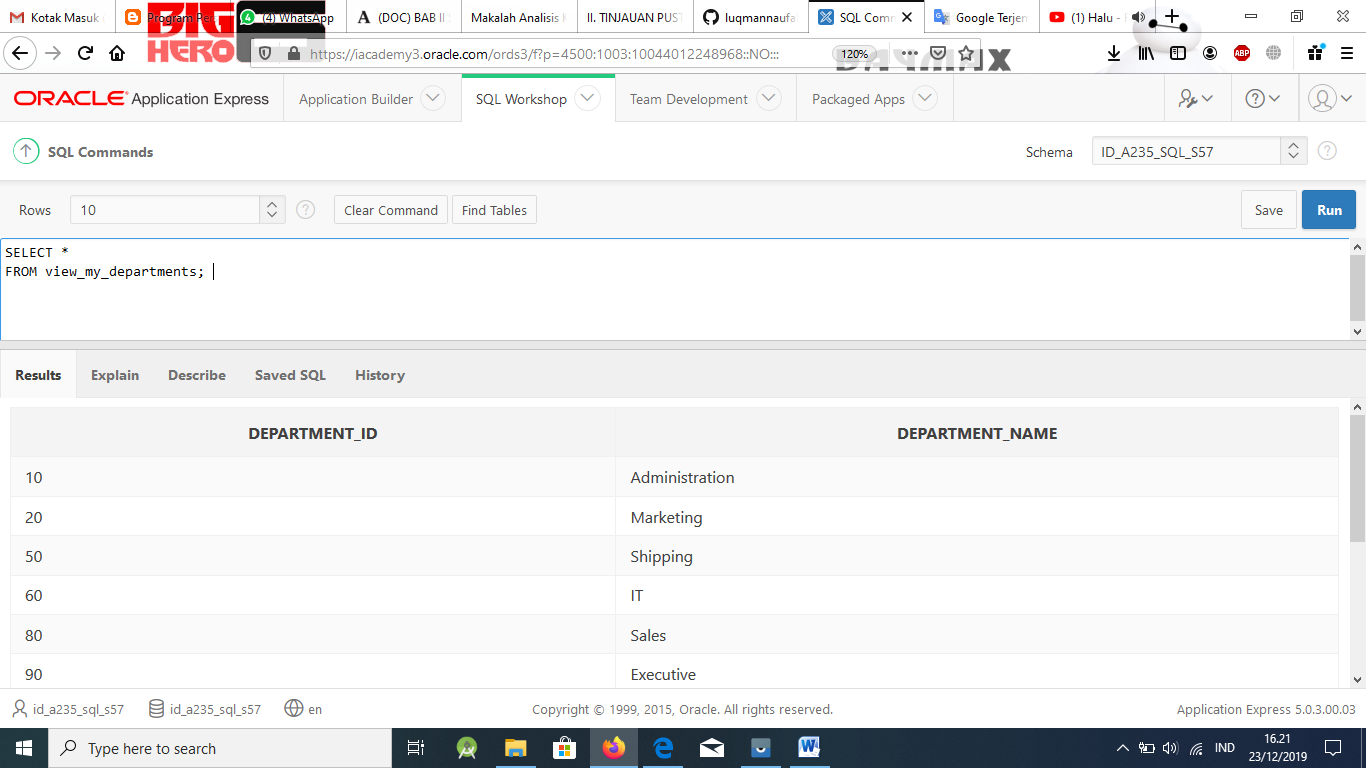


1. Verify that the new additions to my\_departments were added using view\_my\_departments.

### Solution:

SELECT \*

FROM view\_my\_departments; - See chart below

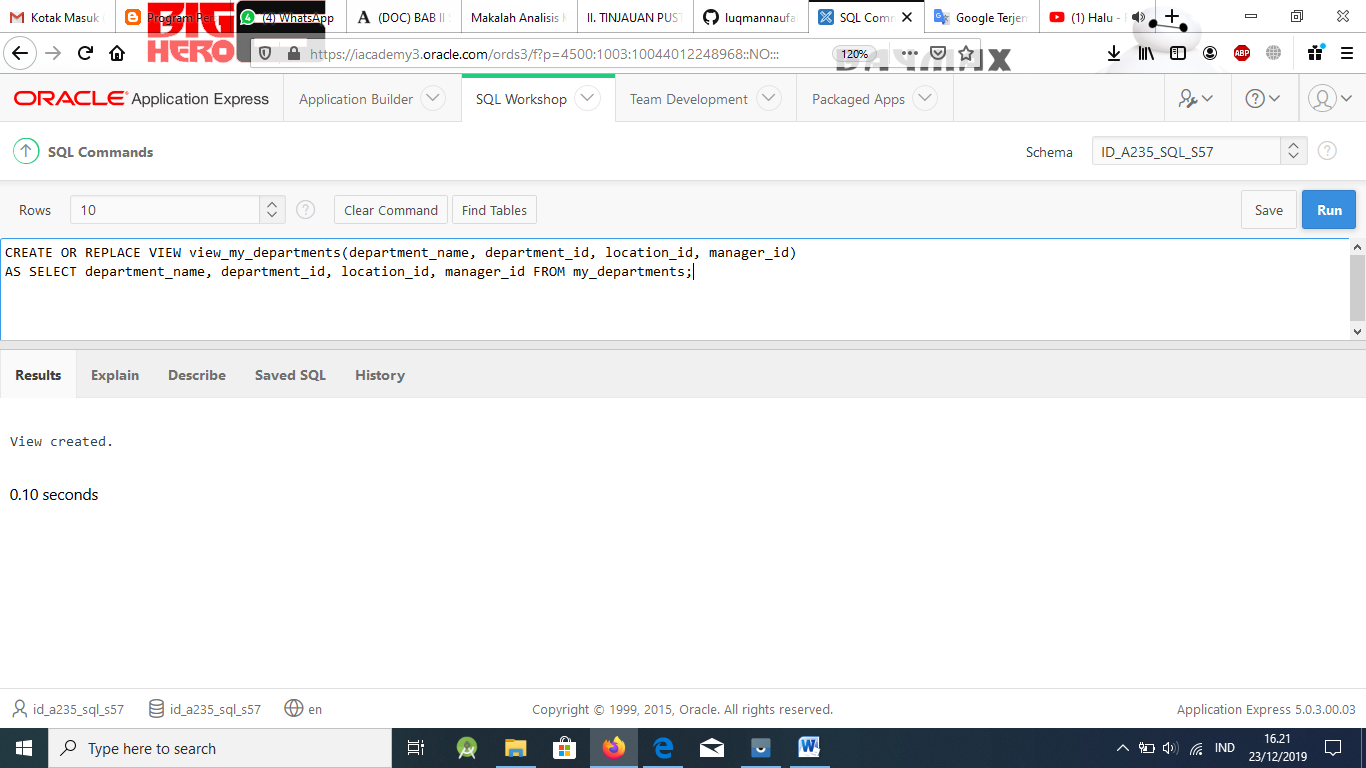


1. Modify view\_my\_departments to include location ID. Do a SELECT \* command to show what columns are present and a DESCRIBE command to view the columns and associat- ed constraints.

### Solution:

CREATE OR REPLACE VIEW view\_my\_departments(department\_name, department\_id, location\_id, manager\_id)

AS SELECT department\_name, department\_id, location\_id, manager\_id FROM my\_departments;



1. Make location\_id a NOT NULL column in the my\_departments table.

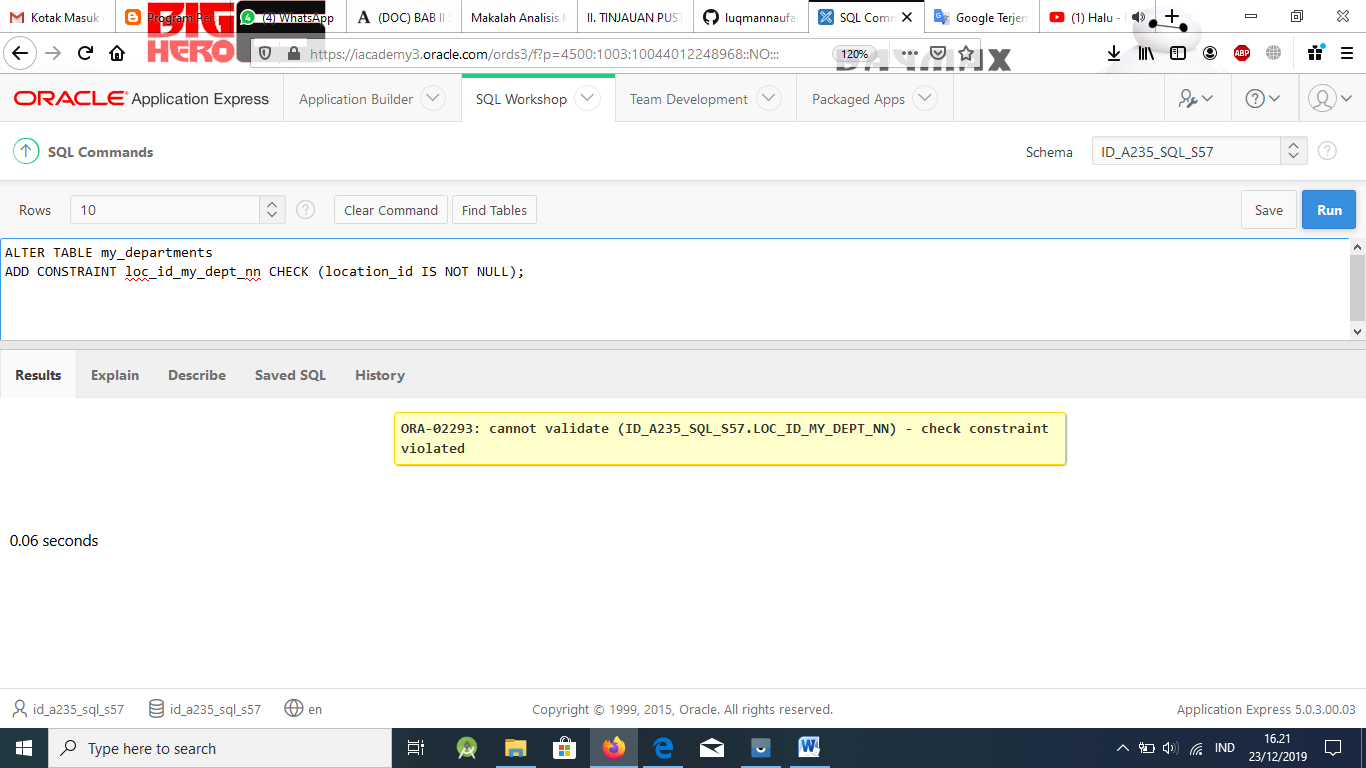
### Solution:

ALTER TABLE my\_departments

ADD CONSTRAINT loc\_id\_my\_dept\_nn CHECK (location\_id IS NOT NULL);

ORA-02293: cannot validate (USCA\_ORACLE\_SQL01\_S01.LOC\_ID\_MY\_DEPT\_NN) - check constraint violated

\*\* adding the three new columns to the my\_departments table inserted null values in the location\_id column. A NOT NULL constraint can only be added to a column if the table is empty or if the column has a value for every row.



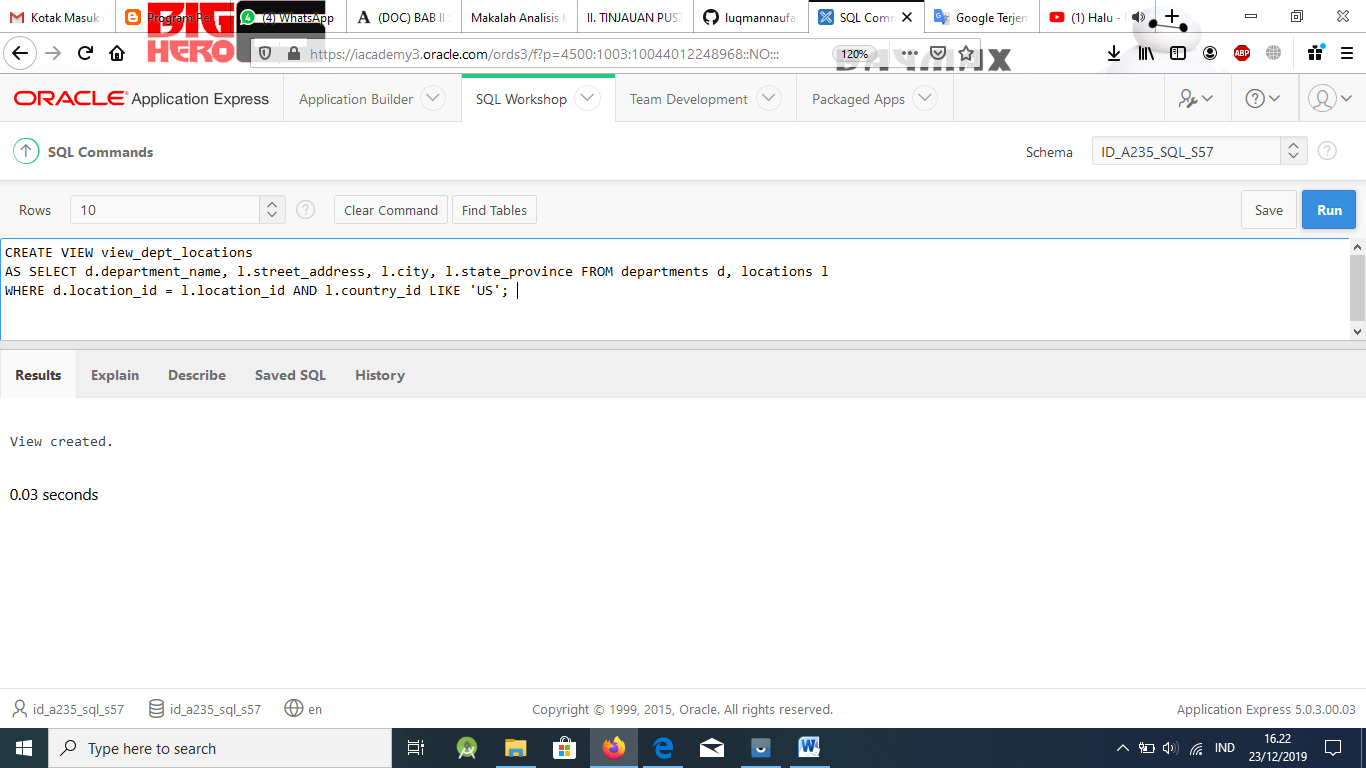
1. Using the Oracle database, create a complex view between locations and departments with only the following columns: department\_name, street\_address, city, and state. In- clude only U.S. cities. Verify that the view was created using a SELECT \* statement.

### Solution:

CREATE VIEW view\_dept\_locations

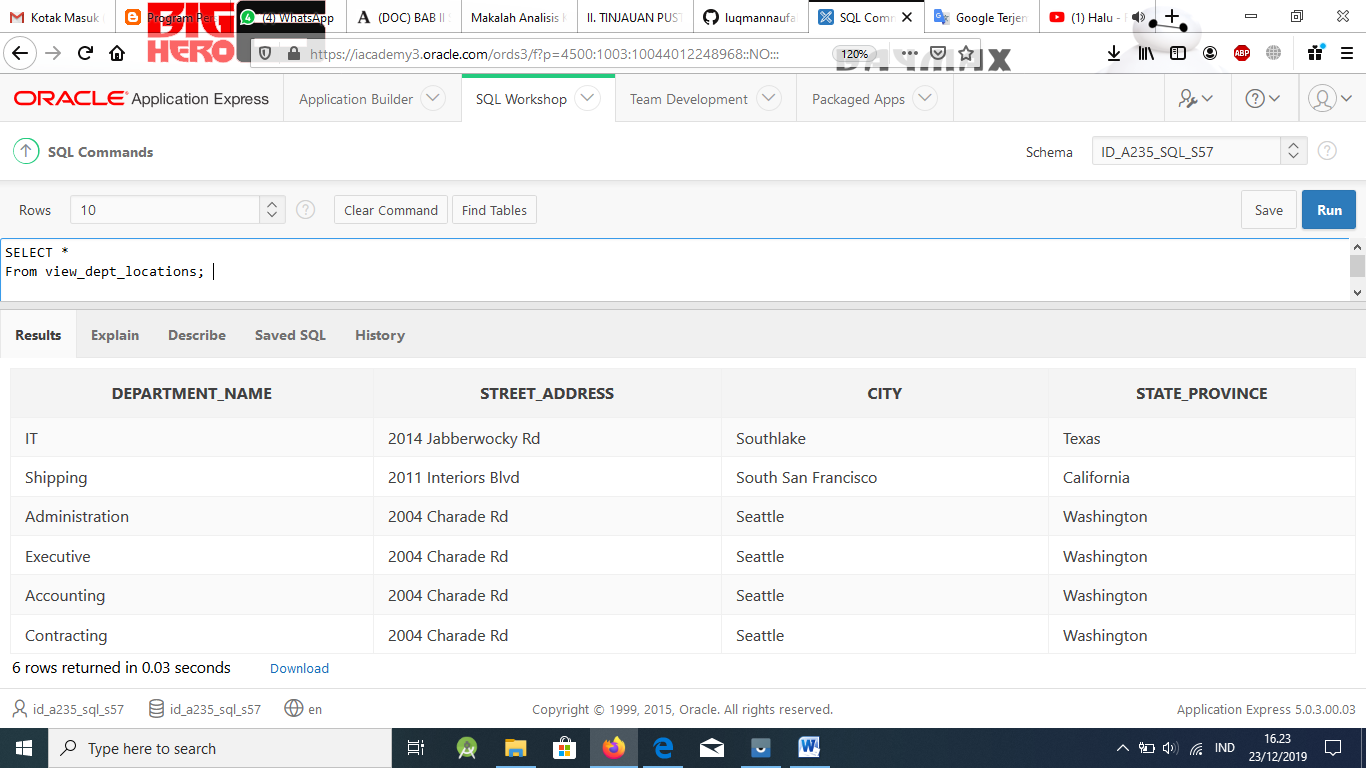
AS SELECT d.department\_name, l.street\_address, l.city, l.state\_province FROM departments d, locations l

WHERE d.location\_id = l.location\_id AND l.country\_id LIKE 'US';



SELECT \*

From view\_dept\_locations;



See chart below

|  |  |  |  |
| --- | --- | --- | --- |
| **DEPARTMENT\_ID** | **DEPART- MENT\_NAME** | **MANAGER\_ID** | **LOCATION\_ID** |
| 10 | Administration | 200 | 1700 |
| 20 | Marketing | 201 | 1800 |
| 50 | Shipping | 124 | 1500 |
| 60 | IT | 103 | 1400 |
| 80 | Sales | 149 | 2500 |
| 90 | Executive | 100 | 1700 |
| 110 | Accounting | 205 | 1700 |
| 190 | Contracting | (null) | 1700 |
| 105 | Advertising | (null) | (null) |
| 120 | Custodial | (null) | (null) |
| 220 | Human Re- sources | (null) | (null) |
| 130 | Planning | (null) | (null) |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| IT | 2014 Jabberwocky Rd | Southlake | Texas |
| Shipping | 2011 Interiors Blvd | South San Francisco | California |
| Administration | 2004 Charade Rd | Seattle | Washington |
| Executive | 2004 Charade Rd | Seattle | Washington |
| Accounting | 2004 Charade Rd | Seattle | Washington |
| Contracting | 2004 Charade Rd | Seattle | Washington |