UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SECD2523 DATABASE

SQL 5a | TCL

LECTURE LEARNING OUTCOME

By the end of this lecture, students should be able to:

Construct SQL statements for

01 **View**

02 **Transaction**

01 View

02 Transaction

Before we continue this TCL,

we reuse the same database same tables created in DML3

If failed to find the database and tables you created previously,

refer here to get our SQL statements to prepare all tables and sample data…
https://docs.google.com/document/d/1jC5daHE4WhP7N9U6ZxqRzBJcJAzqJxf1/edit?usp=sharing&ouid=112935562328060013817&rtpof=true&sd=true

Here, our tables with records

select * from locations;

```
mysql> select * from locations;
+-------------+----------------------------------------+-------------+--------------------+-------------------+------------+
| location_id | street_address                         | postal_code | city               | state_province    | country_id |
+-------------+----------------------------------------+-------------+--------------------+-------------------+------------+
|        1000 | 1297 Via Cola di Rie                   | 00989       | Roma               | NULL              | IT         |
|        1100 | 93091 Calle della Testa                | 10934       | Venice             | NULL              | IT         |
|        1200 | 2017 Shinjuku-ku                       | 1689        | Tokyo              | Tokyo Prefecture  | JP         |
|        1300 | 9450 Kamiya-cho                        | 6823        | Hiroshima          | NULL              | JP         |
|        1400 | 2014 Jabberwocky Rd                    | 26192       | Southlake          | Texas             | US         |
|        1500 | 2011 Interiors Blvd                    | 99236       | South San Francisco | California        | US         |
|        1600 | 2007 Zagora St                         | 50090       | South Brunswick    | New Jersey        | US         |
|        1700 | 2004 Charade Rd                        | 98199       | Seattle            | Washington        | US         |
|        1800 | 147 Spadina Ave                        | M5V 2L7     | Toronto            | Ontario           | CA         |
|        1900 | 6092 Boxwood St                        | YSW 9T2     | Whitehorse         | Yukon             | CA         |
|        2000 | 40-5-12 Laogianggen                    | 190518      | Beijing            | NULL              | CN         |
|        2100 | 1298 Vileparle (E)                     | 490231      | Bombay             | Maharashtra       | IN         |
|        2200 | 12-98 Victoria Street                  | 2901        | Sydney             | New South Wales   | AU         |
|        2300 | 198 Clementi North                     | 540198      | Singapore          | NULL              | SG         |
|        2400 | 8204 Arthur St                         | NULL        | London             | NULL              | UK         |
|        2500 | Magdalen Centre, The Oxford Science Park | OX9 9ZB     | Oxford             | Oxford            | UK         |
|        2600 | 9702 Chester Road                      | 09629850293 | Stretford          | Manchester        | UK         |
|        2700 | Schwanthalerstr. 7031                  | 80925       | Munich             | Bavaria           | DE         |
|        2800 | Rua Frei Caneca 1360                   | 01307-002   | Sao Paulo          | Sao Paulo         | BR         |
|        2900 | 20 Rue des Corps-Saints                | 1730        | Geneva             | Geneve            | CH         |
|        3000 | Murtenstrasse 921                      | 3095        | Bern               | BE                | CH         |
|        3100 | Pieter Breughelstraat 837              | 3029SK      | Utrecht            | Utrecht           | NL         |
|        3200 | Mariano Escobedo 9991                  | 11932       | Mexico City        | Distrito Federal, | MX         |
+-------------+----------------------------------------+-------------+--------------------+-------------------+------------+
23 rows in set (0.00 sec)
```

Here, our tables with records

select * from departments;

```
mysql> select * from departments;
+---------------+----------------------+------------+-------------+
| department_id | department_name      | manager_id | location_id |
+---------------+----------------------+------------+-------------+
|            10 | Administration       |        200 |        1700 |
|            20 | Marketing            |        201 |        1800 |
|            30 | Purchasing           |        114 |        1700 |
|            40 | Human Resources      |        203 |        2400 |
|            50 | Shipping             |        121 |        1500 |
|            60 | IT                   |        103 |        1400 |
|            70 | Public Relations     |        204 |        2700 |
|            80 | Sales                |        145 |        2500 |
|            90 | Executive            |        100 |        1700 |
|           100 | Finance              |        108 |        1700 |
|           110 | Accounting           |        205 |        1700 |
|           120 | Treasury             |       NULL |        1700 |
|           130 | Corporate Tax        |       NULL |        1700 |
|           140 | Control And Credit   |       NULL |        1700 |
|           150 | Shareholder Services |       NULL |        1700 |
|           160 | Benefits             |       NULL |        1700 |
|           170 | Manufacturing        |       NULL |        1700 |
|           180 | Construction         |       NULL |        1700 |
|           190 | Contracting          |       NULL |        1700 |
|           200 | Operations           |       NULL |        1700 |
|           210 | IT Support           |       NULL |        1700 |
|           220 | NOC                  |       NULL |        1700 |
|           230 | IT Helpdesk          |       NULL |        1700 |
|           240 | Government Sales     |       NULL |        1700 |
|           250 | Retail Sales         |       NULL |        1700 |
|           260 | Recruiting           |       NULL |        1700 |
|           270 | Payroll              |       NULL |        1700 |
+---------------+----------------------+------------+-------------+
27 rows in set (0.00 sec)
```

Here,
our tables with records

select * from employees;

```
mysql> select * from employees;
+-------------+------------+-------------+----------+--------------------+------------+------------+----------+----------------+------------+---------------+
| employee_id | first_name | last_name   | email    | phone_number       | hire_date  | job_id     | salary   | commission_pct | manager_id | department_id |
+-------------+------------+-------------+----------+--------------------+------------+------------+----------+----------------+------------+---------------+
|         100 | Steven     | King        | SKING    | 515.123.4567       | 1987-06-17 | AD_PRES    | 24000.00 |           NULL |       NULL |            90 |
|         101 | Neena      | Kochhar     | NKOCHHAR | 515.123.4568       | 1989-09-21 | AD_VP      | 17000.00 |           NULL |        100 |            90 |
|         102 | Lex        | De Haan     | LDEHAAN  | 515.123.4569       | 1993-01-13 | AD_VP      | 17000.00 |           NULL |        100 |            90 |
|         103 | Alexander  | Hunold      | AHUNOLD  | 590.423.4567       | 1990-01-03 | IT_PROG    |  9000.00 |           NULL |        102 |            60 |
|         104 | Bruce      | Ernst       | BERNST   | 590.423.4568       | 1991-05-21 | IT_PROG    |  6000.00 |           NULL |        103 |            60 |
|         105 | David      | Austin      | DAUSTIN  | 590.423.4569       | 1997-06-25 | IT_PROG    |  4800.00 |           NULL |        103 |            60 |
|         106 | Valli      | Pataballa   | VPATABAL | 590.423.4560       | 1998-02-05 | IT_PROG    |  4800.00 |           NULL |        103 |            60 |
|         107 | Diana      | Lorentz     | DLORENTZ | 590.423.5567       | 1999-02-07 | IT_PROG    |  4200.00 |           NULL |        103 |            60 |
|         108 | Nancy      | Greenberg   | NGREENBE | 515.124.4569       | 1994-08-17 | FI_MGR     | 12000.00 |           NULL |        101 |           100 |
|         109 | Daniel     | Faviet      | DFAVIET  | 515.124.4169       | 1994-08-16 | FI_ACCOUNT |  9000.00 |           NULL |        108 |           100 |
|         110 | John       | Chen        | JCHEN    | 515.124.4269       | 1997-09-28 | FI_ACCOUNT |  8200.00 |           NULL |        108 |           100 |
|         111 | Ismael     | Sciarra     | ISCIARRA | 515.124.4369       | 1997-09-30 | FI_ACCOUNT |  7700.00 |           NULL |        108 |           100 |
|         112 | Jose Manuel | Urman      | JMURMAN  | 515.124.4469       | 1998-03-07 | FI_ACCOUNT |  7800.00 |           NULL |        108 |           100 |
|         113 | Luis       | Popp        | LPOPP    | 515.124.4567       | 1999-12-07 | FI_ACCOUNT |  6900.00 |           NULL |        108 |           100 |
|         114 | Den        | Raphaely    | DRAPHEAL | 515.127.4561       | 1994-12-07 | PU_MAN     | 11000.00 |           NULL |        100 |            30 |
|         115 | Alexander  | Khoo        | AKHOO    | 515.127.4562       | 1995-05-18 | PU_CLERK   |  3100.00 |           NULL |        114 |            30 |
|         116 | Shelli     | Baida       | SBAIDA   | 515.127.4563       | 1997-12-24 | PU_CLERK   |  2900.00 |           NULL |        114 |            30 |
|         117 | Sigal      | Tobias      | STOBIAS  | 515.127.4564       | 1997-07-24 | PU_CLERK   |  2800.00 |           NULL |        114 |            30 |
|         118 | Guy        | Himuro      | GHIMURO  | 515.127.4565       | 1998-11-15 | PU_CLERK   |  2600.00 |           NULL |        114 |            30 |
|         119 | Karen      | Colmenares  | KCOLMENA | 515.127.4566       | 1999-08-10 | PU_CLERK   |  2500.00 |           NULL |        114 |            30 |
|         120 | Matthew    | Weiss       | MWEISS   | 650.123.1234       | 1996-07-18 | ST_MAN     |  8000.00 |           NULL |        100 |            50 |
|         121 | Adam       | Fripp       | AFRIPP   | 650.123.2234       | 1997-04-10 | ST_MAN     |  8200.00 |           NULL |        100 |            50 |
|         122 | Payam      | Kaufling    | PKAUFLIN | 650.123.3234       | 1995-05-01 | ST_MAN     |  7900.00 |           NULL |        100 |            50 |
|         123 | Shanta     | Vollman     | SVOLLMAN | 650.123.4234       | 1997-10-10 | ST_MAN     |  6500.00 |           NULL |        100 |            50 |
|         124 | Kevin      | Mourgos     | KMOURGOS | 650.123.5234       | 1999-11-16 | ST_MAN     |  5800.00 |           NULL |        100 |            50 |
|         180 | Winston    | Taylor      | WTAYLOR  | 650.507.9876       | 1998-01-24 | SH_CLERK   |  3200.00 |           NULL |        120 |            50 |
|         181 | Jean       | Fleaur      | JFLEAUR  | 650.507.9877       | 1998-02-23 | SH_CLERK   |  3100.00 |           NULL |        120 |            50 |
|         182 | Martha     | Sullivan    | MSULLIVA | 650.507.9878       | 1999-06-21 | SH_CLERK   |  2500.00 |           NULL |        120 |            50 |
|         183 | Girard     | Geoni       | GGEONI   | 650.507.9879       | 2000-02-03 | SH_CLERK   |  2800.00 |           NULL |        120 |            50 |
|         184 | Nandita    | Sarchand    | NSARCHAN | 650.509.1876       | 1996-01-27 | SH_CLERK   |  4200.00 |           NULL |        121 |            50 |
|         185 | Alexis     | Bull        | ABULL    | 650.509.2876       | 1997-02-20 | SH_CLERK   |  4100.00 |           NULL |        121 |            50 |
|         186 | Julia      | Dellinger   | JDELLING | 650.509.3876       | 1998-06-24 | SH_CLERK   |  3400.00 |           NULL |        121 |            50 |
|         187 | Anthony    | Cabrio      | ACABRIO  | 650.509.4876       | 1999-02-07 | SH_CLERK   |  3000.00 |           NULL |        121 |            50 |
|         188 | Kelly      | Chung       | KCHUNG   | 650.505.1876       | 1997-06-14 | SH_CLERK   |  3800.00 |           NULL |        122 |            50 |
|         189 | Jennifer   | Dilly       | JDILLY   | 650.505.2876       | 1997-08-13 | SH_CLERK   |  3600.00 |           NULL |        122 |            50 |
|         190 | Timothy    | Gates       | TGATES   | 650.505.3876       | 1998-07-11 | SH_CLERK   |  2900.00 |           NULL |        122 |            50 |
|         191 | Randall    | Perkins     | RPERKINS | 650.505.4876       | 1999-12-19 | SH_CLERK   |  2500.00 |           NULL |        122 |            50 |
|         192 | Sarah      | Bell        | SBELL    | 650.501.1876       | 1996-02-04 | SH_CLERK   |  4000.00 |           NULL |        123 |            50 |
|         193 | Britney    | Everett     | BEVERETT | 650.501.2876       | 1997-03-03 | SH_CLERK   |  3900.00 |           NULL |        123 |            50 |
|         194 | Samuel     | McCain      | SMCCAIN  | 650.501.3876       | 1998-07-01 | SH_CLERK   |  3200.00 |           NULL |        123 |            50 |
|         195 | Vance      | Jones       | VJONES   | 650.501.4876       | 1999-03-17 | SH_CLERK   |  2800.00 |           NULL |        123 |            50 |
|         196 | Alana      | Walsh       | AWALSH   | 650.507.9811       | 1998-04-24 | SH_CLERK   |  3100.00 |           NULL |        124 |            50 |
|         197 | Kevin      | Feeney      | KFEENEY  | 650.507.9822       | 1998-05-23 | SH_CLERK   |  3000.00 |           NULL |        124 |            50 |
|         198 | Donald     | OConnell    | DOCONNEL | 650.507.9833       | 1999-06-21 | SH_CLERK   |  2600.00 |           NULL |        124 |            50 |
|         199 | Douglas    | Grant       | DGRANT   | 650.507.9844       | 2000-01-13 | SH_CLERK   |  2600.00 |           NULL |        124 |            50 |
|         200 | Jennifer   | Whalen      | JWHALEN  | 515.123.4444       | 1987-09-17 | AD_ASST    |  4400.00 |           NULL |        101 |            10 |
|         201 | Michael    | Hartstein   | MHARTSTE | 515.123.5555       | 1996-02-17 | MK_MAN     | 13000.00 |           NULL |        100 |            20 |
|         202 | Pat        | Fay         | PFAY     | 603.123.6666       | 1997-08-17 | MK_REP     |  6000.00 |           NULL |        201 |            20 |
|         203 | Susan      | Mavris      | SMAVRIS  | 515.123.7777       | 1994-06-07 | HR_REP     |  6500.00 |           NULL |        101 |            40 |
|         204 | Hermann    | Baer        | HBAER    | 515.123.8888       | 1994-06-07 | PR_REP     | 10000.00 |           NULL |        101 |            70 |
|         205 | Shelley    | Higgins     | SHIGGINS | 515.123.8080       | 1994-06-07 | AC_MGR     | 12000.00 |           NULL |        101 |           110 |
|         206 | William    | Gietz       | WGIETZ   | 51hr5.123.8181     | 1994-06-07 | AC_ACCOUNT |  8300.00 |           NULL |        205 |           110 |
+-------------+------------+-------------+----------+--------------------+------------+------------+----------+----------------+------------+---------------+
107 rows in set (0.00 sec)
```

Let’s start…

View

View

- **Virtual table**
  - is defined by using an SQL SELECT statement to one or more base tables
  - looks like a regular table with named columns and rows of data
  - doesn't store the data itself in the database

- **To create a view**:

  MySQL:

```
CREATE VIEW ViewName [(column_list)] (optional)
AS subselect
[WITH CHECK OPTION] (optional);
```

View

```
CREATE VIEW ViewName [(column_list)] (optional)
AS subselect
[WITH CHECK OPTION] (optional);
```

- A **subselect** is known as the defining query to the view
  - If provide a list of column names, it must match the number of columns produced by the **subselect**
  - If you don't specify column names, the view will automatically use the column names from the **subselect**
  - You must provide column names if there's any confusion or ambiguity, such as:
    - When the **subselect** includes calculated columns without names (e.g., no AS clause was used).
    - When a join produces two columns with the same name.

SELECT * FROM employees;

```
mysql> select * from employees;
+-------------+------------+------------+----------+--------------------+------------+------------+----------+----------------+------------+---------------+
| employee_id | first_name | last_name  | email    | phone_number       | hire_date  | job_id     | salary   | commission_pct | manager_id | department_id |
+-------------+------------+------------+----------+--------------------+------------+------------+----------+----------------+------------+---------------+
|         100 | Steven     | King       | SKING    | 515.123.4567       | 1987-06-17 | AD_PRES    | 24000.00 |           NULL |       NULL |            90 |
|         101 | Neena      | Kochhar    | NKOCHHAR | 515.123.4568       | 1989-09-21 | AD_VP      | 17000.00 |           NULL |        100 |            90 |
|         102 | Lex        | De Haan    | LDEHAAN  | 515.123.4569       | 1993-01-13 | AD_VP      | 17000.00 |           NULL |        100 |            90 |
|         103 | Alexander  | Hunold     | AHUNOLD  | 590.423.4567       | 1990-01-03 | IT_PROG    |  9000.00 |           NULL |        102 |            60 |
|         104 | Bruce      | Ernst      | BERNST   | 590.423.4568       | 1991-05-21 | IT_PROG    |  6000.00 |           NULL |        103 |            60 |
|         105 | David      | Austin     | DAUSTIN  | 590.423.4569       | 1997-06-25 | IT_PROG    |  4800.00 |           NULL |        103 |            60 |
|         106 | Valli      | Pataballa  | VPATABAL | 590.423.4560       | 1998-02-05 | IT_PROG    |  4800.00 |           NULL |        103 |            60 |
|         107 | Diana      | Lorentz    | DLORENTZ | 590.423.5567       | 1999-02-07 | IT_PROG    |  4200.00 |           NULL |        103 |            60 |
```

we are going to focus on department_id = 60 in the following examples…

Create a horizontal view

- Example 1:
  - Create a view so that the head of **department_id=60** can see the details only for staff who work in his or her department.

```
CREATE VIEW dept60_view
AS
SELECT * FROM employees
WHERE department_id = 60;
```

  - Once done created,

    execute the following statement

```
SELECT * FROM dept60_view;
```

    to get the result view

To ensure the head of the department sees only specific rows:

- do not give them access to the base table **employees**.
- instead, give them access to the view **dept60_view**.

```
mysql> create view dept60_view as select * from employees where department_id = 60;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from dept60_view;
+-------------+------------+-----------+----------+--------------+------------+----------+---------+----------------+------------+---------------+
| employee_id | first_name | last_name | email    | phone_number | hire_date  | job_id   | salary  | commission_pct | manager_id | department_id |
+-------------+------------+-----------+----------+--------------+------------+----------+---------+----------------+------------+---------------+
|         103 | Alexander  | Hunold    | AHUNOLD  | 590.423.4567 | 1990-01-03 | IT_PROG  | 9000.00 |           NULL |        102 |            60 |
|         104 | Bruce      | Ernst     | BERNST   | 590.423.4568 | 1991-05-21 | IT_PROG  | 6000.00 |           NULL |        103 |            60 |
|         105 | David      | Austin    | DAUSTIN  | 590.423.4569 | 1997-06-25 | IT_PROG  | 4800.00 |           NULL |        103 |            60 |
|         106 | Valli      | Pataballa | VPATABAL | 590.423.4560 | 1998-02-05 | IT_PROG  | 4800.00 |           NULL |        103 |            60 |
|         107 | Diana      | Lorentz   | DLORENTZ | 590.423.5567 | 1999-02-07 | IT_PROG  | 4200.00 |           NULL |        103 |            60 |
+-------------+------------+-----------+----------+--------------+------------+----------+---------+----------------+------------+---------------+
5 rows in set (0.01 sec)
```

```
mysql> show tables;
+---------------------+
| Tables_in_db_sql_ar |
+---------------------+
| departments         |
| dept60_view         |
| employees           |
| locations           |
+---------------------+
4 rows in set (0.00 sec)
```

Create a vertical view

- Example 2:
  - Create a view of employees' detail at department_id=60 that **exclude salary**, so that only the head of the department can access the salary details for employees who work at their department.

To keep salary details private:
- do not give employees at department_id 60 access to the base table **employees** or **dept60_view**.
- instead, give them access to the view **emp_dept60_view** which hides salary.

```
CREATE VIEW emp_dept60_view
AS
SELECT employee_id, first_name, last_name, email, phone_number, job_id
FROM employees WHERE department_id = 60;
```

or

```
CREATE VIEW emp_dept60_view
AS
SELECT employee_id, first_name, last_name, email, phone_number, job_id
FROM dept60_view;
```

this view is created from the previous slide

```
mysql> select * from emp_dept60_view;
+-------------+------------+-----------+----------+--------------+---------+
| employee_id | first_name | last_name | email    | phone_number | job_id  |
+-------------+------------+-----------+----------+--------------+---------+
|         103 | Alexander  | Hunold    | AHUNOLD  | 590.423.4567 | IT_PROG |
|         104 | Bruce      | Ernst     | BERNST   | 590.423.4568 | IT_PROG |
|         105 | David      | Austin    | DAUSTIN  | 590.423.4569 | IT_PROG |
|         106 | Valli      | Pataballa | VPATABAL | 590.423.4560 | IT_PROG |
|         107 | Diana      | Lorentz   | DLORENTZ | 590.423.5567 | IT_PROG |
+-------------+------------+-----------+----------+--------------+---------+
5 rows in set (0.00 sec)
```

Create a grouped and joined view

- Example 3:
  - Create a view of employees who manage to count of employees working in each department within a specific location.

```
CREATE VIEW DepartmentEmployeeCount
AS
SELECT d.department_name, l.city AS location, COUNT(e.employee_id) AS employee_count
FROM  employees e
JOIN  departments d ON e.department_id = d.department_id
JOIN  locations l ON d.location_id = l.location_id
GROUP BY d.department_name, l.city;
```

Grouped view: use of a **subselect**

Joined view: multiple **tables**

```
mysql> CREATE VIEW DepartmentEmployeeCount
    -> AS
    -> SELECT d.department_name, l.city AS location, COUNT(e.employee_id) AS employee_count
    -> FROM  employees e
    -> JOIN  departments d ON e.department_id = d.department_id
    -> JOIN  locations l ON d.location_id = l.location_id
    -> GROUP BY d.department_name, l.city;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM DepartmentEmployeeCount;
+-----------------+---------------------+----------------+
| department_name | location            | employee_count |
+-----------------+---------------------+----------------+
| Administration  | Seattle             |              1 |
| Marketing       | Toronto             |              2 |
| Purchasing      | Seattle             |              6 |
| Human Resources | London              |              1 |
| Shipping        | South San Francisco |             45 |
| IT              | Southlake           |              5 |
| Public Relations| Munich              |              1 |
| Sales           | Oxford              |             34 |
| Executive       | Seattle             |              3 |
| Finance         | Seattle             |              6 |
| Accounting      | Seattle             |              2 |
+-----------------+---------------------+----------------+
11 rows in set (0.00 sec)
```

- Once done created, execute the following statement

```
SELECT * FROM DepartmentEmployeeCount;
```

to get the result view

Removing a View (DROP VIEW)

- **To remove a view** from the database:

  MySQL: **DROP VIEW** ViewName [**RESTRICT | CASCADE**] (optional);

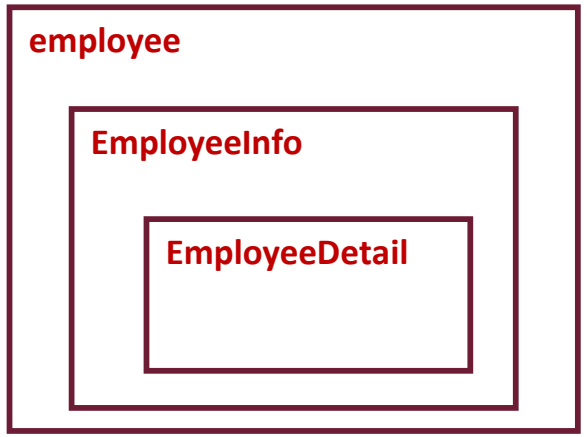
  - If **CASCADE** is specified, **DROP VIEW** will delete the view and all related objects that depend on it. This means any views that are based on the view you are dropping will also be deleted.
  - If **RESTRICT** is specified, the command will only work if there are no dependent objects. If any object depends on the view, the command will be rejected.
  - Note: The default setting is **RESTRICT**.

- Example 4:

  EmployeeInfo view select basic employee information from **employee** table,
  EmployeeDetail view select a part information from **EmployeeInfo** view.

  DROP VIEW EmployeeInfo CASCADE;
  it will delete **EmployeeInfo** and also remove **EmployeeDetail**
  since it depends on **EmployeeInfo**.

  DROP VIEW EmployeeInfo RESTRICT;
  it will **fail** and the **view won't be dropped** since **EmployeeDetail** still depends on **EmployeeInfo**.



View - WITH CHECK OPTION

- **CREATE VIEW … WITH CHECK OPTION**
  - prohibits a row from migrating out of the view
  - **WITH LOCAL CHECK OPTION**:
    any row inserted or updated in the view (or any view that depends on it) must remain visible in the view after the operation. If a row is inserted or updated, it cannot be removed from the view unless it is also removed from the underlying table or view that the original view is based on.
  - **WITH CASCADED CHECK OPTION**:
    default setting and ensures that any row inserted or updated into a view, and any views directly or indirectly defined on it
    (a row must satisfy all conditions in the chain of dependent views, not just the immediate view where the operation occurs. If any condition fails, the operation is rejected.)
- MySQL:

```
CREATE VIEW ViewName [(column_list)] (optional)
AS subselect
[WITH (LOCAL/CASCADED) CHECK OPTION] (optional);
```

View - WITH CHECK OPTION

- Example 5:
  - Create a view with check option so that the head of **department_id=60** can see the details only for staff who work in his or her department.

```
CREATE VIEW dept60_vco
AS SELECT * FROM employees
WHERE department_id = 60
WITH CHECK OPTION;
```

```
mysql> CREATE VIEW dept60_vco AS SELECT * FROM employees WHERE department_id = 60 WITH CHECK OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from dept60_vco;
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
| employee_id | first_name | last_name | email    | phone_number | hire_date  | job_id   | salary   | commission_pct | manager_id | department_id |
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
|         103 | Alexander  | Hunold    | AHUNOLD  | 590.423.4567 | 1990-01-03 | IT_PROG  | 9000.00  |           NULL |        102 |            60 |
|         104 | Bruce      | Ernst     | BERNST   | 590.423.4568 | 1991-05-21 | IT_PROG  | 6000.00  |           NULL |        103 |            60 |
|         105 | David      | Austin    | DAUSTIN  | 590.423.4569 | 1997-06-25 | IT_PROG  | 4800.00  |           NULL |        103 |            60 |
|         106 | Valli      | Pataballa | VPATABAL | 590.423.4560 | 1998-02-05 | IT_PROG  | 4800.00  |           NULL |        103 |            60 |
|         107 | Diana      | Lorentz   | DLORENTZ | 590.423.5567 | 1999-02-07 | IT_PROG  | 4200.00  |           NULL |        103 |            60 |
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
5 rows in set (0.00 sec)
```

  - Once done created,

now attempt to insert new values
(99,'Mary', 'Mohd', 'MARYMD', '590.423.1111',
'1999-02-28', 'IT_PROG', 5300, null, 103, 60)

```
mysql> INSERT INTO dept60_vco values (99,'Mary', 'Mohd', 'MARYMD', '590.423.1111','1999-02-28', 'IT_PROG', 5300, null, 103, 60);
Query OK, 1 row affected (0.00 sec)

mysql> select * from dept60_vco;
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
| employee_id | first_name | last_name | email    | phone_number | hire_date  | job_id   | salary   | commission_pct | manager_id | department_id |
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
|          99 | Mary       | Mohd      | MARYMD   | 590.423.1111 | 1999-02-28 | IT_PROG  | 5300.00  |           NULL |        103 |            60 |
|         103 | Alexander  | Hunold    | AHUNOLD  | 590.423.4567 | 1990-01-03 | IT_PROG  | 9000.00  |           NULL |        102 |            60 |
|         104 | Bruce      | Ernst     | BERNST   | 590.423.4568 | 1991-05-21 | IT_PROG  | 6000.00  |           NULL |        103 |            60 |
|         105 | David      | Austin    | DAUSTIN  | 590.423.4569 | 1997-06-25 | IT_PROG  | 4800.00  |           NULL |        103 |            60 |
|         106 | Valli      | Pataballa | VPATABAL | 590.423.4560 | 1998-02-05 | IT_PROG  | 4800.00  |           NULL |        103 |            60 |
|         107 | Diana      | Lorentz   | DLORENTZ | 590.423.5567 | 1999-02-07 | IT_PROG  | 4200.00  |           NULL |        103 |            60 |
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
6 rows in set (0.00 sec)
```

```
INSERT INTO dept60_vco VALUES (99,'Mary', 'Mohd', 'MARYMD', '590.423.1111','1999-02-28',
'IT_PROG', 5300, null, 103, 60);
```

now attempt to update
the salary for employee_id = 105 from 4800 to 5000

```
mysql> UPDATE dept60_vco SET salary=5000 WHERE employee_id = 105;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from dept60_vco;
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
| employee_id | first_name | last_name | email    | phone_number | hire_date  | job_id   | salary   | commission_pct | manager_id | department_id |
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
|          99 | Mary       | Mohd      | MARYMD   | 590.423.1111 | 1999-02-28 | IT_PROG  | 5300.00  |           NULL |        103 |            60 |
|         103 | Alexander  | Hunold    | AHUNOLD  | 590.423.4567 | 1990-01-03 | IT_PROG  | 9000.00  |           NULL |        102 |            60 |
|         104 | Bruce      | Ernst     | BERNST   | 590.423.4568 | 1991-05-21 | IT_PROG  | 6000.00  |           NULL |        103 |            60 |
|         105 | David      | Austin    | DAUSTIN  | 590.423.4569 | 1997-06-25 | IT_PROG  | 5000.00  |           NULL |        103 |            60 |
|         106 | Valli      | Pataballa | VPATABAL | 590.423.4560 | 1998-02-05 | IT_PROG  | 4800.00  |           NULL |        103 |            60 |
|         107 | Diana      | Lorentz   | DLORENTZ | 590.423.5567 | 1999-02-07 | IT_PROG  | 4200.00  |           NULL |        103 |            60 |
+-------------+------------+-----------+----------+--------------+------------+----------+----------+----------------+------------+---------------+
6 rows in set (0.00 sec)
```

```
UPDATE dept60_vco SET salary = 5000 WHERE employee_id = 105;
```

Transaction – related to topic 7

Transaction

- **COMMIT**
  - ends the transaction successfully
  - making the database changes permanent
  - A new transaction starts after **COMMIT** with the next transaction-initiating statement.
- **ROLLBACK**
  - aborts the transaction
  - backing out any changes made by the transaction
  - A new transaction starts after **ROLLBACK** with the next transaction-initiating statement.
- MySQL:

```
SET TRANSACTION [TRANSACTION MODE];
START TRANSACTION;
[SQL Statements];
ROLLBACK or ROLLBACK TO ...;
COMMIT;
```

COMMIT / ROLLBACK

Example 1:

- Create Table for bank_accounts with account_id, account_name, and balance, then insert sample data before we proceed applying commit or rollback

```
CREATE TABLE bank_accounts (
    account_id INT PRIMARY KEY,
    account_name VARCHAR(50),
    balance DECIMAL(10,2)
);

INSERT INTO bank_accounts VALUES
(70001001, 'Ben Bob', 4020.00),
(70001020, 'Melvin Alex', 1020.30),
(70301030, 'Diana Ernst', 30303.03);
```

```
mysql> select * from bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  4020.00 |
|   70001020 | Melvin Alex  |  1020.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

continue to next slide

COMMIT / ROLLBACK

Example 1 (continued…):

- Now, transfer 100.00

  from account_id = 70001001 to account_id = 70001020

```
START TRANSACTION;

UPDATE bank_accounts
SET balance = balance - 100.00
WHERE account_id = 70001001;

UPDATE bank_accounts
SET balance = balance + 100.00
WHERE account_id = 70001020;

SELECT * FROM bank_accounts;

COMMIT;

SELECT * FROM bank_accounts;
```

Before Transfer

```
mysql> select * from bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  4020.00 |
|   70001020 | Melvin Alex  |  1020.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

```
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> UPDATE bank_accounts
    -> SET balance = balance - 100.00
    -> WHERE account_id = 70001001;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE bank_accounts
    -> SET balance = balance + 100.00
    -> WHERE account_id = 70001020;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3920.00 |
|   70001020 | Melvin Alex  |  1120.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)

mysql>
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3920.00 |
|   70001020 | Melvin Alex  |  1120.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

COMMIT / ROLLBACK

Example 2:

- Now, attempt to transfer 300.00

  from account_id = 70001001 to account_id = 70301030

  but realize a mistake occurred, and want to cancel it.

```
START TRANSACTION;

UPDATE bank_accounts
SET balance = balance - 300.00
WHERE account_id = 70001001;

UPDATE bank_accounts
SET balance = balance + 300.00
WHERE account_id = 70301030;

ROLLBACK;

SELECT * FROM bank_accounts;
```

Before Attempt to Transfer

```
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3920.00 |
|   70001020 | Melvin Alex  |  1120.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

```
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> UPDATE bank_accounts
    -> SET balance = balance - 300.00
    -> WHERE account_id = 70001001;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE bank_accounts
    -> SET balance = balance + 300.00
    -> WHERE account_id = 70301030;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3920.00 |
|   70001020 | Melvin Alex  |  1120.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

SAVEPOINT

Example 3:

- Now, attempt to transfer 500.00
  from account_id = 70001001 to account_id = 70001020,

  then a fee of 50.00 is attempted to deduct
  from account_id = 70001020, but something goes wrong.

```
START TRANSACTION;
UPDATE bank_accounts
SET balance = balance - 500.00
WHERE account_id = 70001001;
SAVEPOINT after_deduct_bob;
UPDATE bank_accounts
SET balance = balance + 500.00
WHERE account_id = 70001020;
UPDATE bank_accounts
SET balance = balance - 50.00
WHERE account_id = 70001020;
ROLLBACK TO after_deduct_bob;
COMMIT;
SELECT * FROM bank_accounts;
```

Before Attempt to Transfer

```
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3920.00 |
|   70001020 | Melvin Alex  |  1120.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

```
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE bank_accounts
    -> SET balance = balance - 500.00
    -> WHERE account_id = 70001001;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SAVEPOINT after_deduct_bob;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE bank_accounts
    -> SET balance = balance + 500.00
    -> WHERE account_id = 70001020;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE bank_accounts
    -> SET balance = balance - 50.00
    -> WHERE account_id = 70001020;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> ROLLBACK TO after_deduct_bob;
Query OK, 0 rows affected (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3420.00 |
|   70001020 | Melvin Alex  |  1120.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

Summary

Command	Purpose
COMMIT	Saves all changes made during the transaction.
ROLLBACK	Undoes all changes made during the transaction, reverting to the start.
SAVEPOINT	Creates a checkpoint within the transaction to partially roll back changes.

Transaction Modes

- **READ ONLY**
  - ensures no changes can be made to the database within that transaction
  - restrict write operations (like INSERT, UPDATE, or DELETE)
  - retrieve data only
- **READ WRITE**
  - (default) allows both reading and modifying data
- MySQL:

```
SET TRANSACTION [READ ONLY | READ WRITE];
START TRANSACTION;
[SQL Statements];
ROLLBACK or ROLLBACK TO ...;
COMMIT;
```

SET TRANSACTION [READ ONLY | READ WRITE];

Example 4

- attempt to transfer 500.00 out
  but the transaction mode sets to READ ONLY

```
SET TRANSACTION READ ONLY;

START TRANSACTION;

SELECT * FROM bank_accounts;

UPDATE bank_accounts
SET balance = balance - 500.00
WHERE account_id = 70001001;

COMMIT;
```

```
mysql> SET TRANSACTION READ ONLY;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3420.00 |
|   70001020 | Melvin Alex  |  1120.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)

mysql> UPDATE bank_accounts
    -> SET balance = balance - 500.00
    -> WHERE account_id = 70001001;
ERROR 1792 (25006): Cannot execute statement in a READ ONLY transaction.
mysql>
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3420.00 |
|   70001020 | Melvin Alex  |  1120.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

SET TRANSACTION [READ ONLY | READ WRITE];

Example 5

- attempt to transfer 500.00 from account_id = 70001020 to account_id = 70301030

```
SET TRANSACTION READ WRITE;

START TRANSACTION;

UPDATE bank_accounts
SET balance = balance - 500.00 WHERE account_id = 70001020;

UPDATE bank_accounts
SET balance = balance + 500.00 WHERE account_id = 70301030;

COMMIT;

SELECT * FROM bank_accounts;
```

Before Attempt to Transfer

```
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3420.00 |
|   70001020 | Melvin Alex  |  1120.30 |
|   70301030 | Diana Ernst  | 30303.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

```
mysql> SET TRANSACTION READ WRITE;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> UPDATE bank_accounts
    -> SET balance = balance - 500.00 WHERE account_id = 70001020;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE bank_accounts
    -> SET balance = balance + 500.00 WHERE account_id = 70301030;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3420.00 |
|   70001020 | Melvin Alex  |   620.30 |
|   70301030 | Diana Ernst  | 30803.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

Transaction Modes

- **ISOLATION LEVEL READ UNCOMMITTED**
  - allows reading uncommitted changes from other transactions
- **ISOLATION LEVEL READ COMMITTED**
  - ensures that only committed changes are visible (no reading uncommitted changes)
- **ISOLATION LEVEL REPEATABLE READ**
  - (default) ensures that data read within a transaction remains consistent, even if other transactions modify the data
- **ISOLATION LEVEL SERIALIZABLE**
  - ensures full isolation by locking the rows, preventing other transactions from accessing them until the transaction is complete
- MySQL:

```
SET TRANSACTION [ISOLATION LEVEL READ UNCOMMITTED | READ COMMITTED | REPEATABLE READ |
SERIALIZABLE];
START TRANSACTION;
[SQL Statements];
ROLLBACK or ROLLBACK TO ...;
COMMIT;
```

SET TRANSACTION [ISOLATION LEVEL READ UNCOMMITTED | READ COMMITTED | REPEATABLE READ | SERIALIZABLE];

Example 6

- balance of account_id = 70001001 is temporarily updated to deduct 200 in the database. However, this transaction has not been committed yet, this change is still uncommitted and might not be finalized.

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

START TRANSACTION;

UPDATE bank_accounts
SET balance = balance - 200.00
WHERE account_id = 70001001;

ROLLBACK;

SELECT * FROM bank_accounts;
```

Before transaction

```
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3420.00 |
|   70001020 | Melvin Alex  |   620.30 |
|   70301030 | Diana Ernst  | 30803.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

```
mysql> SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> UPDATE bank_accounts SET balance = balance - 200.00 WHERE account_id = 70001001;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3420.00 |
|   70001020 | Melvin Alex  |   620.30 |
|   70301030 | Diana Ernst  | 30803.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

SET TRANSACTION [ISOLATION LEVEL READ UNCOMMITTED | READ COMMITTED | REPEATABLE READ | SERIALIZABLE];

Example 7

- Transaction 1: balance of account_id = 70001001 is updated to deduct 200 in the database and commit changes.
- Transaction 2: can see the committed changes from Transaction 1.

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
START TRANSACTION;
UPDATE bank_accounts
SET balance = balance - 200.00
WHERE account_id = 70001001;
COMMIT;
```

Before transaction

```
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3420.00 |
|   70001020 | Melvin Alex  |   620.30 |
|   70301030 | Diana Ernst  | 30803.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
START TRANSACTION;
SELECT * FROM bank_accounts;
COMMIT;
```

```
mysql> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE bank_accounts SET balance = balance - 200.00 WHERE account_id = 70001001;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3220.00 |
|   70001020 | Melvin Alex  |   620.30 |
|   70301030 | Diana Ernst  | 30803.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

SET TRANSACTION [ISOLATION LEVEL READ UNCOMMITTED | READ COMMITTED | REPEATABLE READ | SERIALIZABLE];

Example 8

- Transaction 1: view balance of account_id = 70001001 and commit under REPEATABLE READ.
- Transaction 2: update balance of account_id = 700010(deduct 200 in the database

```
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
START TRANSACTION;
SELECT * FROM bank_accounts WHERE account_id = 70001001;
COMMIT;

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
START TRANSACTION;
UPDATE bank_accounts
SET balance = balance - 200.00
WHERE account_id = 70001001;
COMMIT;

SELECT * FROM bank_accounts;
```

Before transaction

```
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3220.00 |
|   70001020 | Melvin Alex  |   620.30 |
|   70301030 | Diana Ernst  | 30803.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

```
mysql> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM bank_accounts WHERE account_id = 70001001;
+------------+--------------+---------+
| account_id | account_name | balance |
+------------+--------------+---------+
|   70001001 | Ben Bob      | 3220.00 |
+------------+--------------+---------+
1 row in set (0.01 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE bank_accounts SET balance = balance - 200.00 WHERE account_id = 70001001;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3020.00 |
|   70001020 | Melvin Alex  |   620.30 |
|   70301030 | Diana Ernst  | 30803.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

SET TRANSACTION [ISOLATION LEVEL READ UNCOMMITTED | READ COMMITTED | REPEATABLE READ | SERIALIZABLE];

Example 9

- Transaction 1: view balance of account_id = 70001001 and commit under SERIALIZABLE.
- Transaction 2: update balance of account_id = 700010(deduct 500 in the database

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
START TRANSACTION;
SELECT * FROM bank_accounts WHERE account_id = 70001001;
COMMIT;

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
START TRANSACTION;
UPDATE bank_accounts
SET balance = balance - 500.00
WHERE account_id = 70001001;
COMMIT;

SELECT * FROM bank_accounts;
```

Before transaction

```
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  3020.00 |
|   70001020 | Melvin Alex  |   620.30 |
|   70301030 | Diana Ernst  | 30803.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

```
mysql> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM bank_accounts WHERE account_id = 70001001;
+------------+--------------+---------+
| account_id | account_name | balance |
+------------+--------------+---------+
|   70001001 | Ben Bob      | 3020.00 |
+------------+--------------+---------+
1 row in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE bank_accounts SET balance = balance - 500.00 WHERE account_id = 70001001;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SELECT * FROM bank_accounts;
+------------+--------------+----------+
| account_id | account_name | balance  |
+------------+--------------+----------+
|   70001001 | Ben Bob      |  2520.00 |
|   70001020 | Melvin Alex  |   620.30 |
|   70301030 | Diana Ernst  | 30803.03 |
+------------+--------------+----------+
3 rows in set (0.00 sec)
```

Summary

Mode	What It Does
READ ONLY	Ensures no data can be modified during the transaction.
READ WRITE	Allows both reading and modifying data during the transaction.
READ UNCOMMITTED	Allows reading uncommitted changes from other transactions (dirty reads).
READ COMMITTED	Prevents dirty reads, but data can still change between reads (non-repeatable reads possible).
REPEATABLE READ	Ensures consistent data for the duration of the transaction (no dirty or non-repeatable reads).
SERIALIZABLE	Ensures complete isolation by locking rows, preventing phantom reads or any concurrent access.

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Innovating Solutions
Menginovasi Penyelesaian