

# Menggunakan SQL dengan MySQL/MariaDB

---

J.620100.020.02



# Pendahuluan

---

# Perkenalan

---

Nama lahir Mohamad Septiawan

Lahir di Jakarta pada 17 September 1972

Tinggal di D'Amerta Residence, Lengkong, Bojongsoang, Kab. Bandung, Jawa Barat  
(<https://goo.gl/maps/wxKiFv98EqJA5mLt7> )

Lulusan Universitas Budi Luhur, Jakarta, S1 – Manajemen Informatika (1998)

Status menikah dengan 3 anak

Bidang kompetensi saat ini adalah pemrograman berbasis mobile communication, database, web, desktop, dan games

# Perkenalan

---

Perkenalkan diri ANDA

- ❖ Nama .....
- ❖ Lahir .....
- ❖ Tinggal .....
- ❖ Pendidikan terakhir .....
- ❖ Kompetensi terakhir .....
- ❖ Pengalaman .....

# Apersepsi

---

- Komunikasi
- Mobile communication
- SO dan Aplikasi
- Keamanan

# SKKNI

---

- Keputusan Menteri Ketenagakerjaan RI mengenai



**MENTERI KETENAGAKERJAAN  
REPUBLIK INDONESIA**

KEPUTUSAN MENTERI KETENAGAKERJAAN  
REPUBLIK INDONESIA  
NOMOR 282 TAHUN 2016  
TENTANG  
PENETAPAN STANDAR KOMPETENSI KERJA NASIONAL INDONESIA  
KATEGORI INFORMASI DAN KOMUNIKASI GOLONGAN POKOK AKTIVITAS  
PEMROGRAMAN, KONSULTASI KOMPUTER DAN KEGIATAN YANG  
BERHUBUNGAN DENGAN ITU (YBDI) BIDANG *SOFTWARE DEVELOPMENT*  
SUBBIDANG PEMROGRAMAN

# SKKNI

---

- Keputusan Menteri Ketenagakerjaan RI (versi web: <https://skkni.kemnaker.go.id/> )

Nomor 282 Tahun 2016

SKKNI Software Development - Pemrograman

---

| Kategori   | Tanggal Dibuat                     | Tanggal Publikasi       |
|--|------------------------------------|-------------------------|
| INFORMASI DAN KOMUNIKASI                                     | Selasa, 8 November 2016            | Selasa, 8 November 2016 |
| Instansi Teknis  | Bidang                             | Jabatan Kerja           |
| Kementerian Komunikasi dan Informatika                       | Software Development - Pemrograman | N/A                     |
| Golongan Pokok   | Keterangan                         |                         |
| AKTIVITAS PEMROGRAMAN, KONSULTASI KOMPUTER DAN KEGIATAN YBDI | Mencabut 2012-615 dan 2005-142     |                         |

# SKKNI

## ■ Unit Kompetensi

**KODE UNIT** : J.620100.020.02

**JUDUL UNIT** : Menggunakan SQL

**DESKRIPSI UNIT**: Unit kompetensi ini berhubungan dengan sikap, pengetahuan, dan keterampilan yang dibutuhkan untuk menggunakan perangkat aplikasi berbasis data deskripsi atau SQL (*Structured Query Language*) pada aspek DML-*Data Manipulation Language*.

| ELEMEN KOMPETENSI  | KRITERIA UNJUK KERJA  |
|--|---|
| 1. Mempersiapkan perangkat lunak aplikasi data deskripsi/SQL | 1.1 Perangkat lunak aplikasi SQL telah dipasang.<br>1.2 Perangkat lunak aplikasi SQL dijalankan.  |
| 2. Menggunakan fitur aplikasi SQL                            | 2.1 Fitur pengolahan DML diidentifikasi.<br>2.2 Fitur pengolahan DML dieksekusi sesuai kebutuhan.   |
| 3. Mengisi tabel   | 3.1 Tabel diisi data menggunakan perintah DML.<br>3.2 Indeks dibangkitkan.<br>3.3 View tabel dibentuk sesuai kebutuhan.   |
| 4. Melakukan operasi relasional                              | 4.1 Fitur pengolahan DML diidentifikasi.<br>4.2 Perintah DML dipergunakan untuk manipulasi antar tabel.<br>4.3 Perintah DML dipergunakan untuk manipulasi antar-view.<br>4.4 Perintah DML ditulis secara efisien. |
| 5. Membuat <i>stored procedure</i>                           | 5.1 <i>Stored Procedure</i> dibuat dengan perintah SQL.<br>5.2 Prosedur diuji diperiksa <i>input</i> dan <i>output</i> -nya.  |
| 6. Membuat <i>function</i>                                   | 6.1 <i>Function</i> dibuat dengan perintah SQL.<br>6.2 Perintah SQL pada <i>function</i> ditulis secara efisien.  |
| 7. Membuat <i>trigger</i>                                    | 7.1 <i>Trigger</i> didefinisikan dengan perintah SQL.<br>7.2 Kesesuaian hasil <i>trigger</i> diuji.   |
| 8. Melakukan perintah <i>commit</i> dan <i>rollback</i>      | 8.1 Perubahan data dengan perintah <i>commit</i> dilakukan.<br>8.2 Pembatalan penulisan data dilakukan dengan <i>rollback</i> .   |



# Tujuan dan manfaat

---

- Dasar
  - Undang-Undang Nomor 19 Tahun 2016
  - Peraturan Pemerintah Nomor 71 Tahun 2019
- Tujuan bagi peserta pelatihan adalah kompeten dalam
  - Pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam menggunakan SW SQL pada DML
- Manfaat bagi peserta pelatihan adalah kompeten dalam
  - Mempersiapkan dan instalasi SW SQL dan menggunakan fiturnya
  - Membuat table dan mengisi record dengan DML serta menambahkan index dan view
  - Melakukan operasi relasional
  - Membuat stored procedure, function, trigger, dan commit/rollback

# Mempersiapkan perangkat lunak aplikasi data deskripsi/SQL

---

ELEMEN 1

# SW aplikasi SQL (MySQL)

---

MySQL dibuat pertama kali oleh My SQL AB dari Swedia oleh

- Swedes David Axmark
- Allan Larsson (Pengembang asli MySQL)
- Finland Swede Michael Widenius (Pengembang asli MySQL)

MySQL ditulis dengan C/C++ tahun 1994 oleh pengembang asli MySQL

Versi awal → 1995 untuk pengguna pribadi

Versi Windows → 1998 untuk Windows 95/NT

Sun Microsystems mengakuisisi perusahaan MySQL AB tahun 2008

Oracle Corp mengakuisisi perusahaan Sun Microsystems tahun 2010 dan saat itu pula Widenius merilis MariaDB

# SW aplikasi SQL (MySQL)

---

Versi 5.5 → fitur storage engine default InnoDB (sebelumnya MyISAM)

Versi 5.6 → fitur query optimizer, NoSQL, tipe data TIMESTAMP

Versi 5.7 → fitur tipe data JSON

Versi 8.0 → fitur NoSQL dan JSON ditingkatkan

Tahun 2019 MySQL menjadi TOP RANK di DB-Engines

Distribusi → gratis (community edition) dan berbayar (enterprise edition)

# SW aplikasi SQL (MySQL Workbench)

---

Merupakan IDE khusus untuk MySQL

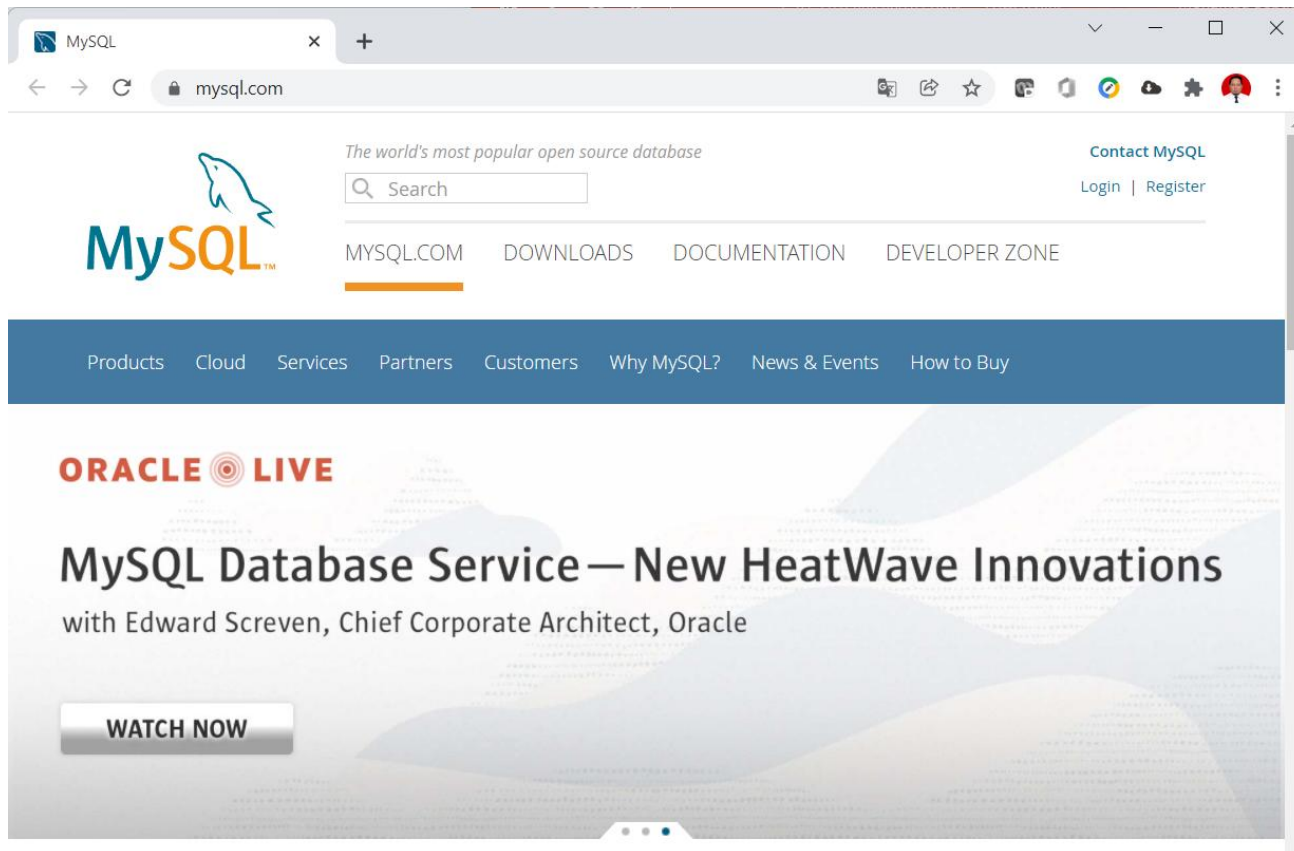
Dibuat oleh MySQL AB dan telah disempurnakan oleh Oracle

Berguna untuk development, administration, design, dan maintenance ke dalam lingkungan tunggal system database MySQL

Versi stabil → 8 berisi SQL editor, Data modelling, Data administration, Performance monitoring, dan Database migration

# Menjalankan SW SQL

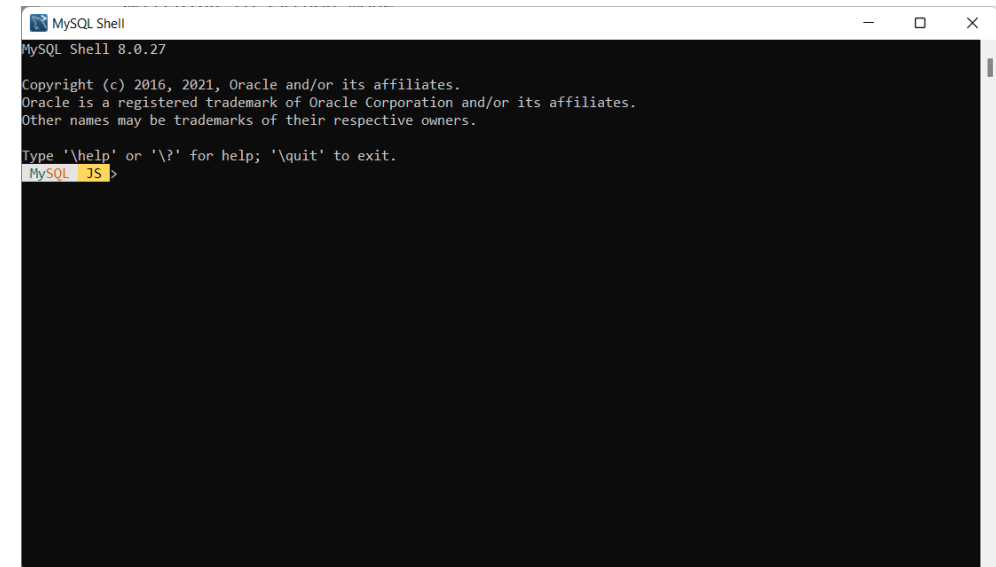
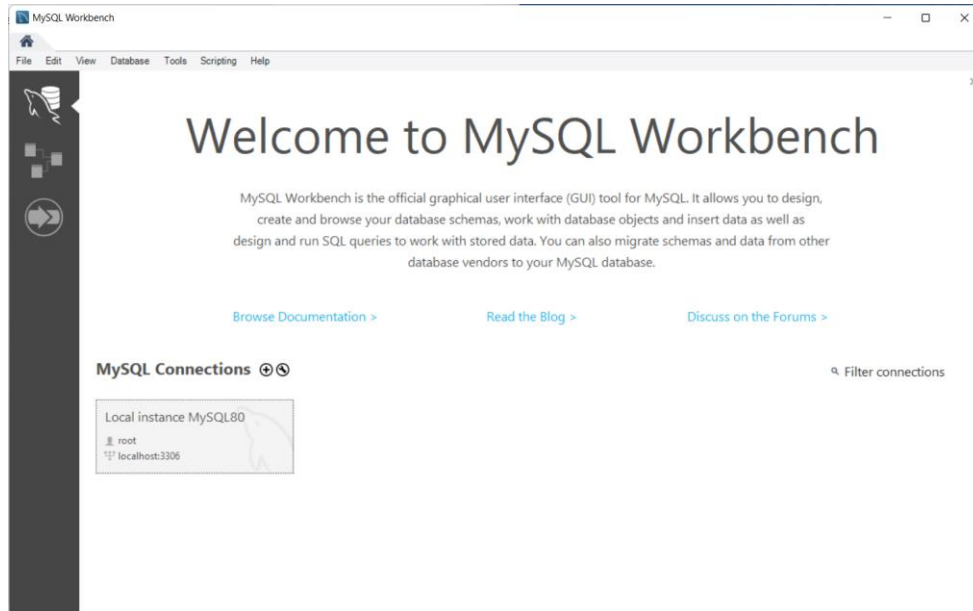
Persiapkan PC dan SW pada PC Anda dan kKunjungi web site MySQL: <https://www.mysql.com/>



# Menjalankan SW SQL

## Mode koneksi

- MySQL Shell
- MySQL Workbench



# Fitur SW aplikasi SQL

---

ELEMEN 2



# Fitur DML

---

SQL kepanjangan dari Structure Query Language

SQL merupakan Bahasa pemrograman untuk RDBMS

SQL didasarkan pada aljabar dan calculus relasional

SQL menggunakan model relasional yang dibuat oleh Edgar F. Codd tahun 1970

SQL memiliki standar ANSI (1986) dan ISO (1987)

SQL pertama kali diaplikasikan pada computer IBM dengan database nya adalah DB2 (1983) dan menyusul Relational Software (sekarang Oracle) membuat aplikasi database Oracle

# Fitur DML

---

Penyebutan database untuk masing-masing vendor menjadi berbeda-beda

- ❑ Microsoft → SQL Server → (Transact) T-SQL
- ❑ Oracle → MySQL → SQL/PSM (Persistent Stored Module)
- ❑ Oracle → Oracle → (Procedural Language) PL/SQL

Berisi tipe perintah yang di klasifikasikan

- DQL: Data Query Language
- DDL: Data Definition Language
- DML: Data Manipulation Language
- DCL: Data Control Language

# Fitur DML

---

Database: kumpulan data terorganisir yang disimpan dan diakses secara elektronik

Istilah lain database:

- ❑ Data: fakta/bukti/kenyataan berupa potongan informasi dalam bentuk nilai kuantitas atau kualitas pada suatu obyek
- ❑ Information: data yang telah diproses, terorganisasi, dan terstruktur. Istilah data dan informasi digunakan secara bergantian
- ❑ Database Management System (DBMS): merupakan software aplikasi database dimana pengguna dapat mengontrol data
- ❑ Relational Database Management System (RDBMS): DBMS yang berbasiskan model relasi yang digunakan untuk keuangan, manufaktur dan logistic, SDM, dan lainnya

# Fitur DML

---

Table/relation: kumpulan data yang disimpan dalam format table yang terdiri dari kolom dan baris sebagai implementasi dari array/vector

Istilah dalam table

❑ Relationship: relasi antar table berdasarkan key yang dimiliki masing-masing table

# Fitur DML

---

Field/Column: kumpulan nilai dari tipe data tertentu dan ukuran nya dapat diatur

Istilah dalam field

- ❑ Tipe data: dapat dikategorikan secara umum → berupa angka atau karakter atau yang lebih rumit seperti gambar
- ❑ Key: factor penentu dalam manipulasi data. Sifat masing-masing key berbeda seperti Primary Key, Secondary/Foreign Key, Alternate Key, Candidate Key, dan sebagainya

# Fitur DML

---

Record/Row: struktur data paling dasar. Merupakan kumpulan kolom-kolom dengan tipe datanya setiap barisnya

Istilah dalam record

- ❑ DML: record yang dapat dimanipulasi (Insert, Update, Delete, dan Select) oleh pengguna dengan perintah SQL
- ❑ Insert: memasukkan record kedalam table pada database
- ❑ Update: memperbaiki record table pada database
- ❑ Delete: menghapus record table pada database
- ❑ Select: mencari/menampilkan data table menggunakan penyaringan, pengurutan, operasi perhitungan/logika pada database

# Fitur DML

---

## Perintah DML pada MySQL

### ❖ Insert

INSERT INTO nama\_table (nama\_field, ... ) VALUES (nilai\_field, ... )

### ❖ Update

UPDATE nama\_table SET nama\_field = nilai\_field, ... WHERE kondisi

### ❖ Delete

DELETE FROM nama\_table WHERE kondisi

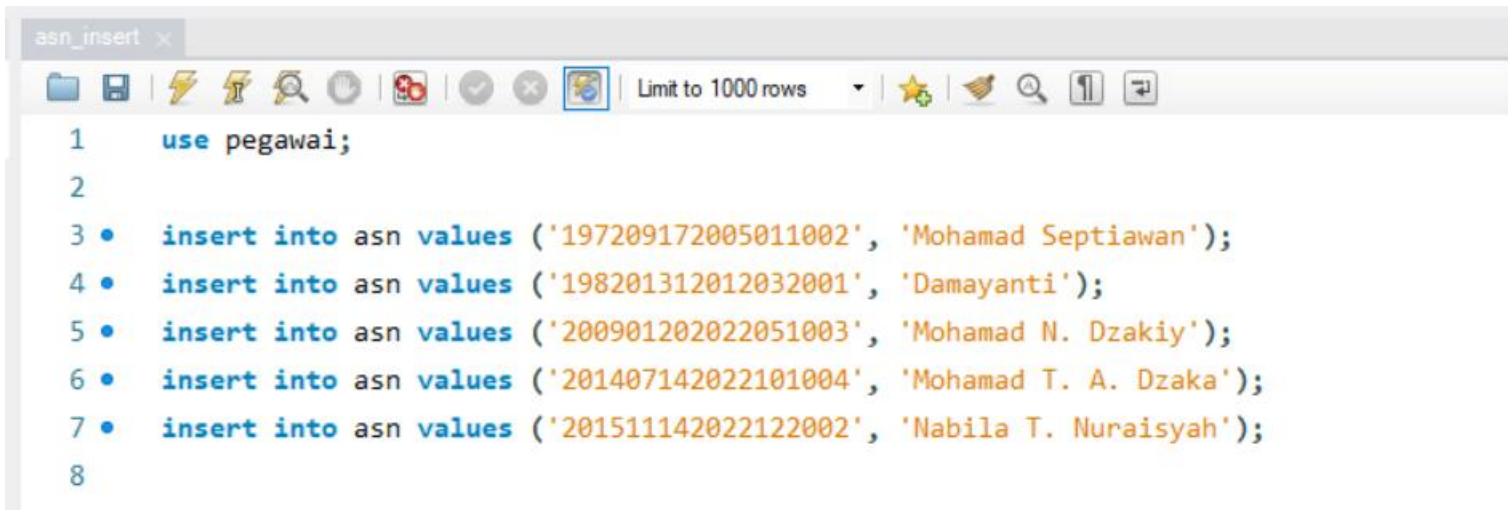
### ❖ Select

SELECT nama\_field, ... FROM nama\_table WHERE kondisi

# Menjalankan DML

---

Tuliskan kode berikut dan eksekusi untuk memasukkan record table



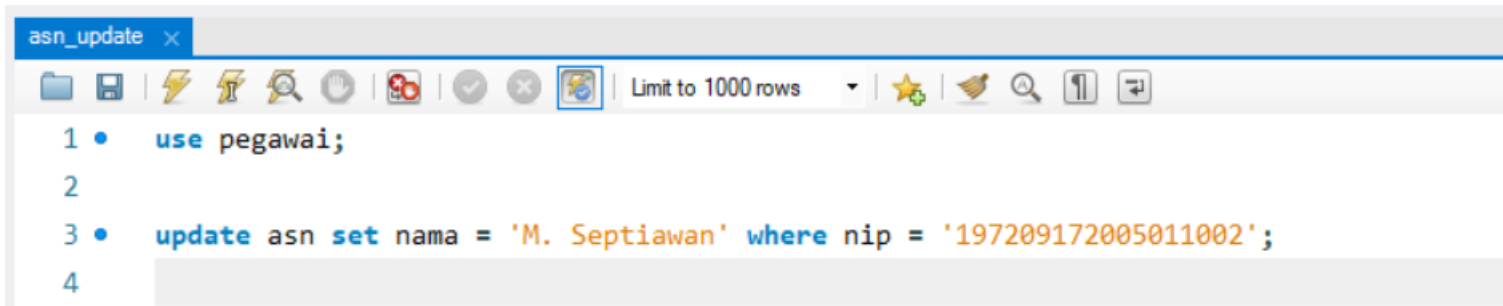
```
asn_insert x
Limit to 1000 rows
1  use pegawai;
2
3 •  insert into asn values ('197209172005011002', 'Mohamad Septiawan');
4 •  insert into asn values ('198201312012032001', 'Damayanti');
5 •  insert into asn values ('200901202022051003', 'Mohamad N. Dzakiy');
6 •  insert into asn values ('201407142022101004', 'Mohamad T. A. Dzaka');
7 •  insert into asn values ('201511142022122002', 'Nabila T. Nuraisyah');
8
```



# Menjalankan DML

---

Tuliskan kode berikut dan eksekusi untuk memperbaiki record table

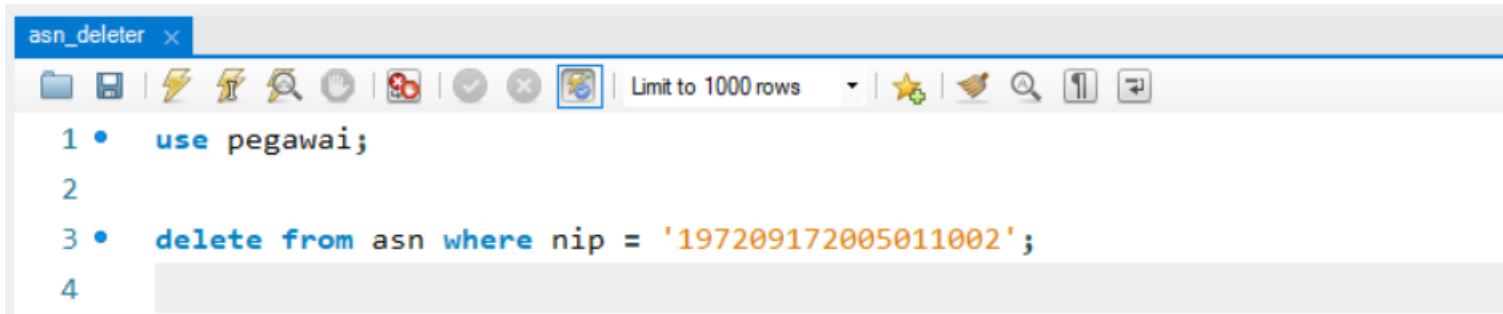


```
asn_update x
1 • use pegawai;
2
3 • update asn set nama = 'M. Septiawan' where nip = '197209172005011002';
4
```

# Menjalankan DML

---

Tuliskan kode berikut dan eksekusi untuk menghapus record table

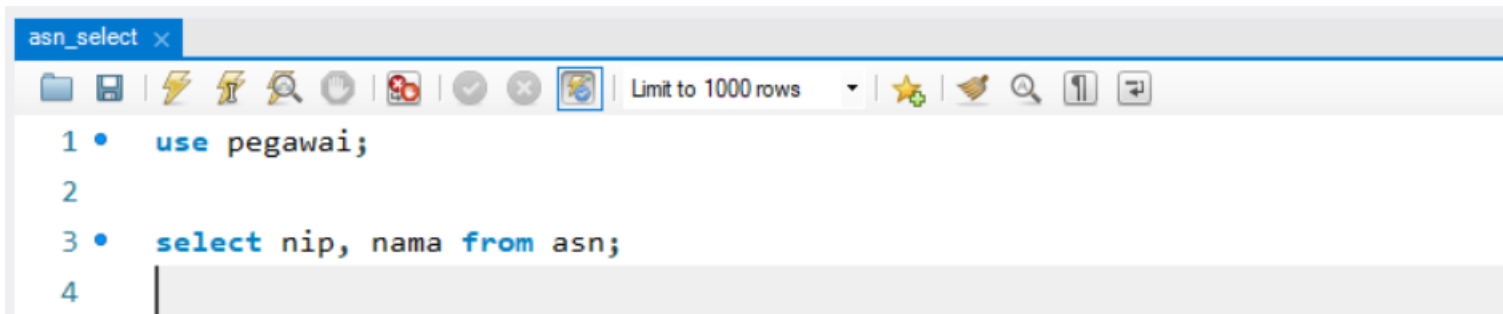


```
asn_deleter x
Limit to 1000 rows
1 • use pegawai;
2
3 • delete from asn where nip = '197209172005011002';
4
```

# Menjalankan DML

---

Tuliskan kode berikut dan eksekusi untuk menampilkan record table

A screenshot of a SQL IDE window titled 'asn\_select'. The window has a toolbar with various icons including a folder, save, lightning bolt, magnifying glass, refresh, redo, undo, and a 'Limit to 1000 rows' dropdown. The main text area contains the following SQL code:

```
1 • use pegawai;  
2  
3 • select nip, nama from asn;  
4
```

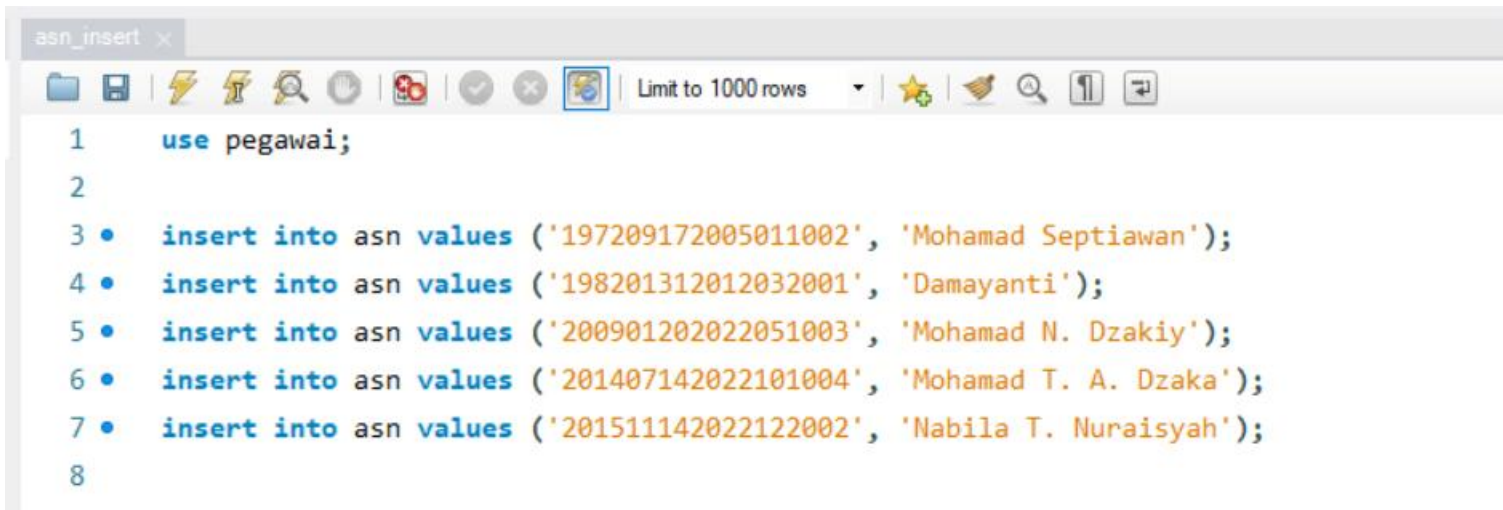
# Mengisi tabel

---

# Mengisi table dengan DML

---

Perintah mengisi record table dengan perintah SQL



```
asn_insert x
Limit to 1000 rows
1  use pegawai;
2
3 • insert into asn values ('197209172005011002', 'Mohamad Septiawan');
4 • insert into asn values ('198201312012032001', 'Damayanti');
5 • insert into asn values ('200901202022051003', 'Mohamad N. Dzakiy');
6 • insert into asn values ('201407142022101004', 'Mohamad T. A. Dzaka');
7 • insert into asn values ('201511142022122002', 'Nabila T. Nuraisyah');
8
```

# Index

---

Digunakan untuk meningkatkan kinerja pada perintah SELECT, dengan menambahkan index pada 1/lebih kolom table

Menyimpang copy nilai dari kolom yang di-index dengan operator tertentu pada klausa WHERE

Index MySQL (Primary Key, Unique, Index, dan FullText) tersimpan dalam B-Trees

Jumlah maksimal index pertabel adalah 16 dan Panjang index 256 byte

# Index

---

Yang perlu diperhatikan dalam membuat index

- ☐ Gunakan tipe data angka yang paling kecil jika memungkinkan untuk menghasilkan 25% table dengan kapasitas kecil
- ☐ Deklarasikan kolom dengan NOT NULL
- ☐ Deklarasikan kolom ke beberapa table jika informasi tersebut sama dan gunakan klausa JOIN
- ☐ Berikan nama index kurang dari 18 karakter
- ☐ Cobalah untuk melakukan normalisasi hingga 3NF
- ☐ Kecepatan lebih penting dari simpanan dan perawatan

# Index

---

Operasi yang menggunakan index

- Mencari baris-baris dengan mencocokkan perintah WHERE secara cepat
- Jika table memiliki banyak kolom index, awalan yang paling kiri yang akan digunakan
- Pencarian nilai terkecil dan terbesar
- Pengurutan atau pengelompokan
- Penggunaan Primary Key



# Index

---

Perintah SQL untuk membuat index

```
CREATE INDEX nama_index ON nama_table (nama_kolom)
```

Perintah SQL untuk menghapus index

```
DROP INDEX nama_index ON nama_table
```

# View

---

Merupakan obyek database, tetapi bukan merupakan database fisik

Merupakan table relasi

Sering juga disebut sebagai table virtual karena hanya berisi perintah SQL yang datanya sabagai referensi dari table fisik

Sering juga dipakai bersamaan dengan stored procedure dan operator untuk memanipulasinya

# View

---

Keuntungan:

- ❖ Tampilan nya dapat mewakili table fisik
- ❖ Tampilan dapat digabungkan dengan table dan atau view lain
- ❖ Tampilan nya dapat bertindak sebagai agregat
- ❖ Tampilan nya dapat menyembunyikan kerumitan data
- ❖ Tampilan nya hanya membutuhkan sedikit ruang penyimpanan
- ❖ Tampilan nya juga dapat memberikan keamanan ekstra
- ❖ Tampilan nya dapat sebagai read-only atau updatable (CRUD pada table fisik)

# View

---

Perintah SQL untuk membuat view

```
CREATE VIEW nama_view AS perintah_select
```

Perintah SQL untuk menampilkan view

```
SELECT kolom FROM views
```

# View

---

Perintah SQL untuk memperbaiki view

```
ALTER VIEW nama_view AS perintah_select
```

Perintah SQL untuk menghapus view

```
DROP VIEW nama_index
```

# Melakukan operasi relasional

---

# Operasi relasi dengan DML

---

## Tipe data

- ❖ Merupakan atribut data yang memberi tahu compiler/interpreter bagaimana data digunakan
- ❖ Untuk membatasi nilai pada variable/fungsi yang digunakan
- ❖ Mendefinisikan operasi yang dapat dilakukan pada data
- ❖ Digunakan dengan tipe system, dengan berbagai cara untuk mendefinisikannya, implementasi, dan bagaimana mempergunakannya
- ❖ Perbedaan tipe system, maka akan berbeda terminology yang digunakannya

# Operasi relasi dengan DML

---

Tipe data MySQL

- ❖ Numeric: Integer, Fixed-Point, Floating-Point, Bit-Value
- ❖ Date dan Time: DATE, DATETIME, TIMESTAMP, TIME, YEAR
- ❖ String: CHAR, VARCHAR, BINARY, VARBINARY, BLOB, TEXT, ENUM, SET



# Operasi relasi dengan DML

---

- ❖ Integer: INTEGER, INT, SMALLINT, TINYINT, MEDIUMINT, BIGINT
  - ❖ INTEGER = INT
  - ❖ TINYINT = BOOL/BOOLEAN → 0: FALSE, 1: TRUE, 2: BUKAN 1 ATAU 0
  - ❖ Atribut: SIGNED, UNSIGNED, AUTO\_INCREMENT, ZEROFILL

# Operasi relasi dengan DML

---

Tabel simpanan dan jangkauan

| Type      | Storage<br>(Bytes) | Minimum<br>Value Signed | Minimum<br>Value<br>Unsigned | Maximum<br>Value Signed | Maximum<br>Value<br>Unsigned |
|-----------|--------------------|-------------------------|------------------------------|-------------------------|------------------------------|
| TINYINT   | 1                  | -128                    | 0                            | 127                     | 255                          |
| SMALLINT  | 2                  | -32768                  | 0                            | 32767                   | 65535                        |
| MEDIUMINT | 3                  | -8388608                | 0                            | 8388607                 | 16777215                     |
| INT       | 4                  | -2147483648             | 0                            | 2147483647              | 4294967295                   |
| BIGINT    | 8                  | $-2^{63}$               | 0                            | $2^{63}-1$              | $2^{64}-1$                   |

# Operasi relasi dengan DML

---

- ❖ Fixed-Point: DECIMAL, NUMERIC
  - ❖ DECIMAL = DEC/FIXED
  - ❖ Memiliki nilai decimal maks. 30 digit
  - ❖ Biasanya digunakan untuk mata uang
- ❖ Floating-Point: FLOAT, DOUBLE
  - ❖ DOUBLE = DOUBLE PRECISION = REAL
- ❖ Bit-Value: BIT
  - ❖ Menyimpan nilai bit dengan jangkauan 1 – 64
  - ❖ Hanya memiliki nilai 1 dan 0

# Operasi relasi dengan DML

---

## ❖ Date dan Time:

- ❖ DATE, DATETIME, TIMESTAMP, TIME, YEAR
- ❖ Merupakan tipe data sementara
- ❖ TIMESTAMP dan DATETIME memiliki tingkah laku otomatis update
- ❖ Nilai data dengan notasi string
- ❖ Penulisan penanggalan 'YYYY-MM-DD' contoh: '19720917' atau '1972-09-17'
- ❖ Penulisan waktu 'hh:mm:ss'

# Operasi relasi dengan DML

---

## ❖ String

- ❖ CHAR, VARCHAR, BINARY, VARBINARY, BLOB, TEXT, ENUM, SET
- ❖ BINARY = CHAR
- ❖ VARBINARY = VARCHAR
- ❖ CHAR dan VARCHAR merupakan tipe yang sama, hanya berbeda cara penyimpanannya

| Value      | CHAR ( 4 ) | Storage Required | VARCHAR ( 4 ) | Storage Required |
|------------|------------|------------------|---------------|------------------|
| ' '        | '    '     | 4 bytes          | ' '           | 1 byte           |
| 'ab'       | 'ab  '     | 4 bytes          | 'ab'          | 3 bytes          |
| 'abcd'     | 'abcd'     | 4 bytes          | 'abcd'        | 5 bytes          |
| 'abcdefgh' | 'abcd'     | 4 bytes          | 'abcd'        | 5 bytes          |

# Operasi relasi dengan DML

---

## ❖ String

- ❖ BLOB memiliki TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB
- ❖ TEXT memiliki TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT
- ❖ BLOB diperlakukan sebagai string biner
- ❖ TEXT diperlakukan bukan sebagai string biner
- ❖ ENUM merupakan string obyek yang memiliki pilihan nilai yang diijinkan dengan index dimulai dari 1
- ❖ ENUM memiliki maks. 65535 elemen

| Value     | Index |
|-----------|-------|
| NULL      | NULL  |
| ' '       | 0     |
| 'Mercury' | 1     |
| 'Venus '  | 2     |
| 'Earth '  | 3     |

# Operasi relasi dengan DML

---

- ❖ String

- ❖ SET merupakan obyek string yang dapat memiliki nilai kosong atau lebih

# Operasi relasi dengan DML

---

## Operator

### ➤ Operator prioritas

```
INTERVAL
BINARY, COLLATE
!
- (unary minus), ~ (unary bit inversion)
^
*, /, DIV, %, MOD
-, +
<<, >>
&
|
= (comparison), <=>, >=, >, <=, <, <>, !=, IS, LIKE, REGEXP, IN, MEMBER OF
BETWEEN, CASE, WHEN, THEN, ELSE
NOT
AND, &&
XOR
OR, ||
= (assignment), :=
```



# Operasi relasi dengan DML

## Operator

➤ Operator pembandingan

= > < >= <= <> !=

➤ Operator logika

| Name    | Description   |
|---------|---------------|
| AND, && | Logical AND   |
| NOT, !  | Negates value |
| OR,     | Logical OR    |
| XOR     | Logical XOR   |

➤ Operator assignment

| Name | Description  |
|------|--|
| :=   | Assign a value   |
| =    | Assign a value (as part of a <a href="#">SET</a> statement, or as part of the <a href="#">SET</a> clause in an <a href="#">UPDATE</a> statement) |

# Operasi relasi dengan DML

---

## Operator

### ➤ Operator aritmetika

| Name                              | Description                     |
|-----------------------------------|---------------------------------|
| <code>%</code> , <code>MOD</code> | Modulo operator                 |
| <code>*</code>                    | Multiplication operator         |
| <code>+</code>                    | Addition operator               |
| <code>-</code>                    | Minus operator                  |
| <code>-</code>                    | Change the sign of the argument |
| <code>/</code>                    | Division operator               |
| <code>DIV</code>                  | Integer division                |

### ➤ Operator pembandingan untuk string

| Name                    | Description                         |
|-------------------------|-------------------------------------|
| <code>LIKE</code>       | Simple pattern matching             |
| <code>NOT LIKE</code>   | Negation of simple pattern matching |
| <code>STRCMP ( )</code> | Compare two strings                 |

# Operasi relasi dengan DML

---

## Fungsi pengendalian alur

- ❑ Kondisi
  - ❑ IF
  - ❑ CASE
- ❑ Perulangan
  - ❑ LOOP
  - ❑ WHILE
  - ❑ REPEAT; menghasilkan string yang berisi string diulang

## Fungsi operasi

- ❖ Matematika
- ❖ Penanggalan
- ❖ String
- ❖ Aggregate

# Manipulasi antar table dengan DML

---

## SELECT

- ❖ Digunakan untuk menampilkan baris yang dipilih dari 1/lebih table/view
- ❖ Berdasarkan kolom-kolom yang sudah ditentukan
- ❖ Gunakan table/view yang diperlukan
- ❖ Gunakan pula filter yang diinginkan
- ❖ Gunakan fungsi dan operator untuk melakukan perhitungan
- ❖ Gunakan klausa tambahan seperti \*, AS,

# Manipulasi antar table dengan DML

## Perintah SELECT

```
SELECT
  [ALL | DISTINCT | DISTINCTROW ]
  [HIGH_PRIORITY]
  [STRAIGHT_JOIN]
  [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
  [SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
  select_expr [, select_expr] ...
  [into_option]
  [FROM table_references
    [PARTITION partition_list]]
  [WHERE where_condition]
  [GROUP BY {col_name | expr | position}, ... [WITH ROLLUP]]
  [HAVING where_condition]
  [WINDOW window_name AS (window_spec)
    [, window_name AS (window_spec)] ...]
  [ORDER BY {col_name | expr | position}
    [ASC | DESC], ... [WITH ROLLUP]]
  [LIMIT {[offset,] row_count | row_count OFFSET offset}]
  [into_option]
  [FOR {UPDATE | SHARE}
    [OF tbl_name [, tbl_name] ...]
    [NOWAIT | SKIP LOCKED]
    | LOCK IN SHARE MODE]
  [into_option]

into_option: {
  INTO OUTFILE 'file_name'
    [CHARACTER SET charset_name]
    export_options
  | INTO DUMPFILE 'file_name'
  | INTO var_name [, var_name] ...
}
```

# Manipulasi antar table dengan DML

---

## UNION

- ❖ Merupakan kombinasi menjadi 1 hasil dari beberapa perintah SELECT
- ❖ Nama kolom yang ditampilkan dari beberapa SELECT, hanya diambil dari SELECT yang pertama

# Manipulasi antar table dengan DML

---

## Perintah UNION

```
SELECT ...  
UNION [ALL | DISTINCT] SELECT ...  
[UNION [ALL | DISTINCT] SELECT ...]
```

# Manipulasi antar table dengan DML

---

## Subquery

- Perintah SELECT dengan perintah lain
- Keuntungan:



# Manipulasi antar view dengan DML

---

Hampir mirip seperti pada table

# Efisiensi perintah DML

---

Efisiensi dapat menggunakan klausa yang diperlukan seperti

- ❑ Mewakili seluruh nama kolom gunakan \*
- ❑ AS untuk membuat alias terhadap nama kolom atau table
- ❑ Gunakan fungsi dan operator yang dibutuhkan untuk memanipulasi record

# Membuat stored procedure

---

ELEMEN 5

# Stored procedure dengan SQL

---

Penyebutan lain → proc/sp/stopro

Merupakan subrutin yang

- ☐ Berisi perintah SQL dan kondisi/perulangan
- ☐ Tersedia pada aplikasi yang mengakses RDBMS
- ☐ Logika terpusat
- ☐ Menghemat waktu dan memori
- ☐ Dapat memproses beberapa perintah sekaligus
- ☐ Bisa tersimpan bersarang
- ☐ Dapat dipanggil berulang-ulang

# Stored procedure dengan SQL

---

## Keuntungan

- Mengurangi kesibukan jaringan: seluruh perintah SQL di kapsulkan (encapsulated) ke dalam stored procedure. Ketika di eksekusi dan dikirim ke query, yang terkirim termasuk nama dan parameternya
- Logika terpusat: dapat digunakan berulang dan mengurangi usaha dalam duplikasi logika yang sama serta lebih konsisten
- Aman: lebih aman karena dapat diatur penggunaanya dan mencegah dari SQL Injection

# Stored procedure dengan SQL

---

## Kerugian

- Penggunaan sumber daya: meningkatnya penggunaan memori dan prosesor setiap koneksi
- Troubleshooting: kesulitan melakukan debugging dan fasilitasnya
- Perawatan: membutuhkan keahlian khusus

# Stored procedure dengan SQL

---

Pemisah/pembatas

- Membutuhkan notasi untuk pernyataan dan eksekusi secara terpisah
- Memperlakukan sebagai 1 perintah SQL, meskipun didalam nya terdapat beberapa perintah SQL yang diakhir dengan notasi semi-colon (baca ;)
- Perintah:  

```
DELIMITER karakter_delimiter  
perintah_SQL karakter_delimiter  
DELIMITER ;
```
- Karakter umum delimiter // atau \$\$

# Stored procedure dengan SQL

---

## Perintah membuat

DELIMITER //

CREATE PROCEDURE sp\_nama (nama\_parameter tipe\_data, ... )

BEGIN

    Perintah\_SQL

END //

DELIMITER ;



# Stored procedure dengan SQL

---

## Perintah memperbaiki

```
DELIMITER //
```

```
ALTER PROCEDURE sp_nama
```

```
BEGIN
```

```
    Perintah_SQL
```

```
END //
```

```
DELIMITER ;
```

# Stored procedure dengan SQL

---

## **Perintah menghapus**

`DROP PROCEDURE IF EXISTS sp_nama`

# Stored procedure dengan SQL

---

Catatan perintah

- CREATE; gunakan IF NOT EXISTS, jika terdapat kesamaan nama
- CREATE; parameter digunakan untuk menguji argumen
- CREATE; mode parameter IN (default), OUT, dan INOUT. Nilai dari IN mode terlindungi hingga stored procedure berakhir. Nilai dari OUT tidak dapat di inisialisasi ketika dieksekusi
- ALTER; tidak dapat merubah parameter dan body secara langsung, jika menginginkan hapus dan buat baru
- DROP; gunakan IF EXISTS untuk mencegah kesalahan bila tidak ditemukan

# Menguji I/O stored procedure

---

## Eksekusi

CALL sp\_nama (nama\_parameter, ... )

## Catatan perintah

- CREATE; gunakan IF NOT EXISTS, jika terdapat kesamaan nama
- ALTER; tidak dapat merubah parameter, jika menginginkan hapus dan buat baru
- DROP; gunakan IF EXISTS untuk mencegah kesalahan bila tidak ditemukan

# Membuat function

---

ELEMEN 6

# Function dengan SQL

---

Bentuk khusus dari pengkapsulan (encapsulation) rumus/aturan bisnis

Mengembalikan nilai tunggal

# Function dengan SQL

---

## Perintah membuat

DELIMITER //

CREATE FUNCTION sf\_nama (nama\_parameter tipe\_data, ... )

RETURNS tipe\_data DETERMINISTIC

BEGIN

    RETURN

    Perintah\_SQL

END //

DELIMITER ;

# Function dengan SQL

---

## Perintah menghapus

DROP FUNCTION IF EXISTS sf\_nama



# Efisiensi perintah function

---

# Membuat trigger

---

ELEMEN 7

# Trigger dengan SQL

---

Stored program yang otomatis dipanggil karena respon dari suatu kejadian seperti INSERT, UPDATE, DELETE yang terjadi pada suatu table

Tipe tingkatan

- Row: dieksekusi untuk @ baris yang dimasukan, diupdate, dan dihapus
- Statement: dieksekusi sekali untuk setiap transaksi terlepas dari berapa banyak baris yang dimasukan, diupdate, dan dihapus

MySQL hanya mendukung Row Level Trigger

# Trigger dengan SQL

---

## Keuntungan

- Cara yang berbeda untuk menguji integritas data
- Penanganan kesalahan dari lapisan database
- Alternatif lain untuk menjalankan penjadwalan secara otomatis dengan aksi BEFORE atau AFTER
- Berguna untuk audit perubahan data pada table

# Trigger dengan SQL

---

Untuk membedakan nilai diantara kolom, aksi BEFORE | AFTER dapat digunakan modifier NEW | OLD

| Trigger Event | OLD | NEW |
|---------------|-----|-----|
| INSERT        | No  | Yes |
| UPDATE        | Yes | Yes |
| DELETE        | Yes | No  |

# Trigger dengan SQL

---

## Perintah membuat

CREATE TRIGGER nama\_trigger

BEFORE | AFTER                  INSERT | UPDATE | DELETE

ON nama\_table FOR EACH ROW

Trigger\_body;

# Trigger dengan SQL

---

## Perintah menghapus

`DROP TRIGGER IF EXISTS nama_trigger`

# Menguji perintah trigger

## Membuat trigger

```
create_trigger_add_record_asn x
1 • use pegawai;
2
3 • create trigger after_nip_ditambah
4   after insert on asn
5   for each row
6   insert into asn_audit
7   set aksi = 'Tambah',
8       nip = new.nip,
9       nama = new.nama,
10      tgl = now();
```

## Hasil dari trigger

| Result Grid  |      |                    |      |        |                     |
|--------------|------|--------------------|------|--------|---------------------|
| Filter Rows: |      |                    |      |        |                     |
|              | id   | nip                | nama | aksi   | tgl                 |
| ▶            | 1    | 197209173005011002 | Dedi | Tambah | 2022-02-20 17:38:24 |
| *            | NULL | NULL               | NULL | NULL   | NULL                |



# Melakukan perintah commit/rollback

---

ELEMEN 8

# Perubahan data pada commit

---

Apabila terjadi kegagalan dalam DML karena beberapa masalah

Untuk memulai transaksi, gunakan perintah `START TRANSACTION`. `BEGIN/BEGIN WORK` merupakan alias nya

Untuk menjalankan transactioin saat ini dan membuat perubahan permanent, gunakan perintah `COMMIT`

Untuk mode mematikan atau menghidupkan transaction saat ini, gunakan perintah `SET AUTOCOMMIT`

# Perubahan data pada rollback

---

Apabila terjadi kegagalan dalam DML karena beberapa masalah

Untuk memulai transaksi, gunakan perintah `START TRANSACTION`. `BEGIN/BEGIN WORK` merupakan alias nya

Untuk mengembalikan transaction saat ini dan membatalkan perubahan, gunakan perintah `ROLLBACK`

Untuk mode mematikan atau menghidupkan transaction saat ini, gunakan perintah `SET AUTOCOMMIT`

# Kesimpulan

---

1. Dapat mempersiapkan dan menjalankan SW SQL yang telah terpasang serta memahami fitur didalamnya
2. Membuat table dan mengisi data menggunakan perintah DML serta membangkitkan index dan view
3. Melakukan operasi relasional seperti manipulasi table dan view
4. Membuat stored procedure, function, trigger, dan commit/rollback

# Pertanyaan 01

---

Dimanakah untuk mendapat aplikasi MySQL yang benar?

Dimanakah untuk mendapat aplikasi MySQL dengan Direct download

Dimanakah untuk mendapat Dokumentasi MySQL

Dimanakah untuk mendapat file pendukung MySQL

# Jawaban 01

---

Dimanakah untuk mendapat aplikasi MySQL yang benar?

<https://www.mysql.com/>

<https://dev.mysql.com/downloads/windows/installer/8.0.html>

<https://www.oracle.com/index.html>

Dimanakah untuk mendapat aplikasi MySQL dengan Direct download

<https://downloads.mysql.com/archives/get/p/25/file/mysql-installer-community-8.0.27.1.msi>

Dimanakah untuk mendapat Dokumentasi MySQL

<https://downloads.mysql.com/docs/refman-8.0-en.a4.pdf>

<https://downloads.mysql.com/docs/workbench-en.a4.pdf>

# Jawaban 01

---

Dimanakah untuk mendapat file pendukung MySQL

<https://www.microsoft.com/en-us/download/details.aspx?id=42642>

<https://docs.microsoft.com/en-US/cpp/windows/latest-supported-vc-redist?view=msvc-170>

# Pertanyaan 02

---

Apakah yang dilakukan sebelum membuat perintah DML

Perintah SQL untuk CRUD



# Jawaban 02

---

Apakah yang dilakukan sebelum membuat perintah DML

**Membuat database dan tablenya terlebih dahulu**

Perintah SQL untuk CRUD

**INSERT INTO nama\_table (nama\_field, ... ) VALUES (nilai\_field, ... )**

**UPDATE nama\_table SET nama\_field = nilai\_field, ... WHERE kondisi**

**DELETE FROM nama\_table WHERE kondisi**

**SELECT nama\_field, ... FROM nama\_table WHERE kondisi**

# Pertanyaan 03

---

Bagaimanakah memasukkan beberapa record sekaligus dalam satu perintah DML

Bagaimanakah membuat index sekaligus dalam membuat table?

Bagaimanakah untuk memanggil view dari view yang sudah ada?

# Jawaban 03

---

Bagaimanakah memasukkan beberapa record sekaligus dalam satu perintah DML

**INSERT INTO nama\_table (nama\_kolom) VALUES (), (), ... ;**

Bagaimanakah membuat index sekaligus dalam membuat table?

**CREATE TABLE nama\_table (nama\_kolom tipe\_data nama\_key);**

Bagaimanakah untuk memanggil view dari view yang sudah ada?

**CREATE VIEW nama\_view AS SELECT nama\_kolom FROM nama\_table;**

**SELECT nama\_kolom FROM nama\_view;**

# Pertanyaan 04

---

Apakah klausa yang digunakan untuk memberi alias pada nama kolom?

Fungsi yang digunakan menentukan kondisi adalah

Bagaimana perintah fungsi IF

Perintah SQL untuk memanipulasi antar table dan atau view adalah

# Pertanyaan 04

---

Apakah klausa yang digunakan untuk memberi alias pada nama kolom?

**Klausa AS**

Fungsi yang digunakan menentukan kondisi

**IF dan CASE**

Bagaimana perintah fungsi IF

**IF (pengujian\_logika, nilainya\_jika\_benar, nilainya\_jika\_salah)**

Perintah SQL untuk memanipulasi antar table dan atau view adalah

**SELECT**

# Pertanyaan 05

---

Mode parameter apakah yang digunakan untuk memasukkan data pada procedure

Apakah kegunaan dari klausa DELIMITER?

Notasi karakter yang digunakan oleh klausa DELIMITER adalah

Perintah yang digunakan untuk memanggil procedure adalah

# Jawaban 05

---

Mode parameter apakah yang digunakan untuk memasukkan data pada procedure

**Mode parameter IN**

Apakah kegunaan dari klausa DELIMITER?

**Membatasi perintah procedure dari perintah SQL lainnya yang berada didalamnya**

Notasi karakter yang digunakan oleh klausa DELIMITER adalah

**Karakter // atau \$\$**

Perintah yang digunakan untuk memanggil procedure adalah

**Klausa CALL**

# Pertanyaan 06

---

Mode parameter apakah yang digunakan untuk memasukkan data pada function

Ciri-ciri dari perintah function yang berbeda dari procedure adalah

Perintah yang digunakan untuk memanggil function adalah



# Jawaban 06

---

Mode parameter apakah yang digunakan untuk memasukkan data pada function

**Tidak ada**

Ciri-ciri dari perintah function yang berbeda dari procedure adalah

**Returns, tipe data, deterministic, return**

Perintah yang digunakan untuk memanggil function adalah

**Perintah SELECT**

# Pertanyaan 07

---

Waktu trigger yang digunakan pada saat trigger event INSERT adalah

Trigger event yang dapat menggunakan waktu trigger AFTER atau BEFORE adalah

# Jawaban 07

---

Waktu trigger yang digunakan pada saat trigger event INSERT adalah

**After**

Trigger event yang dapat menggunakan waktu trigger AFTER atau BEFORE adalah

**Update**

# Pertanyaan 08

---

Perintah yang digunakan untuk memulai transaksi adalah

Perintah untuk menjalankan transaction saat ini dan membuat perubahan permanent

Nilai yang digunakan untuk menghidupkan transaksi adalah

Perintah untuk mengembalikan transaction saat ini dan membatalkan perubahan adalah

# Jawaban 08

---

Perintah yang digunakan untuk memulai transaksi adalah

**START TRANSACTION**

Perintah untuk menjalankan transaction saat ini dan membuat perubahan permanent

**COMMIT**

Nilai yang digunakan untuk menghidupkan transaksi adalah

**AUTO COMMIT = 1**

Perintah untuk mengembalikan transaction saat ini dan membatalkan perubahan adalah

**ROLLBACK**

# Praktek

---

Gunakan jobsheet berikut ini [JS 010 .J.620100.020.02](#) untuk instalasi MySQL

Gunakan jobsheet berikut ini [JS 020 .J.620100.020.02](#) untuk melakukan perintah DML

Gunakan jobsheet berikut ini [JS 032 .J.620100.020.02](#) untuk membuat perintah index

Gunakan jobsheet berikut ini [JS 033 .J.620100.020.02](#) untuk membuat perintah view

Gunakan jobsheet berikut ini [JS 041 .J.620100.020.02](#) untuk membuat operasi relasi

Gunakan jobsheet berikut ini [JS 042 .J.620100.020.02](#) untuk membuat manipulasi antar table

Gunakan jobsheet berikut ini [JS 043 .J.620100.020.02](#) untuk membuat manipulasi antar view

# Praktek

---

Gunakan jobsheet berikut ini [JS 050 .J.620100.020.02](#) untuk membuat procedure

Gunakan jobsheet berikut ini [JS 060 .J.620100.020.02](#) untuk membuat function

Gunakan jobsheet berikut ini [JS 070 .J.620100.020.02](#) untuk membuat trigger

Gunakan jobsheet berikut ini [JS 080 .J.620100.020.02](#) untuk membuat commit/rollback

# Pertanyaan

---

Pengembang MySQL saat ini adalah

Mode koneksi disediakan oleh paket MySQL adalah

Klasifikasi perintah pada SQL adalah

Apakah informasi?



# Pertanyaan

---

Key adalah

Apakah fungsi perintah Update pada DML SQL?

Fungsi index adalah

Fungsi view adalah

# Pertanyaan

---

Klausula yang digunakan untuk memanipulasi 2/lebih pada table/view adalah

Apakah fungsi dari DELIMITER?

Karakter umum yang digunakan pada DELIMITER adalah

Perintah untuk memperbaiki function yang telah dibuat adalah

# Pertanyaan

---

Apa sajakah jenis kejadian yang ada pada TRIGGER?

Perintah untuk mematikan dan menghidupkan transaksi adalah

# Jawaban

---

Pengembang MySQL saat ini adalah

**Oracle Corp.**

Mode koneksi disediakan oleh paket MySQL adalah

**Shell dan Workbench**

Klasifikasi perintah pada SQL adalah

**DQL, DDL, DML, dan DCL**

Apakah informasi?

**Data yang telah diproses, terorganisasi, dan terstruktur**

# Jawaban

---

Key adalah

**Faktor penentu dalam memanipulasi data dan masing-masing key memiliki sifat yang berbeda**

Apakah fungsi perintah Update pada DML SQL?

**Memperbaiki record di table pada database**

Fungsi index adalah

**Meningkatkan kinerja pada perintah SELECT dengan meng-copy pada kolom table**

Fungsi view adalah

**Membuat obyek database berupa table virtual yang berisi perintah SQL sebagai referensi dari table fisik**

# Jawaban

---

Klausu yang digunakan untuk memanipulasi 2/lebih pada table/view adalah

**Where**

Apakah fungsi dari DELIMITER?

**Memperlakukan sebagai 1 perintah pada beberapa perintah SQL**

Karakter umum yang digunakan pada DELIMITER adalah

**// atau \$\$**

Perintah untuk memperbaiki function yang telah dibuat adalah

**Tidak ada**

# Jawaban

---

Apa sajakah jenis kejadian yang ada pada TRIGGER?

**INSERT, UPDATE, dan DELETE**

Perintah untuk mematikan dan menghidupkan transaksi adalah

**AUTOCOMMIT**

# Praktek

---

Buatlah kalkulator sederhana dengan hal sebagai berikut

- ☐ Tidak membutuhkan table apapun
- ☐ Input nilai hanya 2 untuk operasinya
- ☐ Operasi kedua input yang digunakan adalah kali, bagi, tambah, dan kurang
- ☐ Gunakan function untuk melakukan fungsinya



# Tugas

---

Carilah video tutorial

- Mengenai transaksi
  - Penggunaan procedure, function, dan atau trigger
  - Penggunaan table minimal sebanyak 2
- Download video tersebut
- Buatlah aplikasi sesuai video tersebut dan buatlah langkah-langkah kerjanya sebagai dokumentasi pembuatan aplikasi tersebut ke file pengolah kata
- Kumpulkan file aplikasi/software/video dan dokumentasi ke Instruktur bersangkutan