|  |  |  |
| --- | --- | --- |
| **BAHAGIAN PENDIDIKAN DAN LATIHAN TEKNIK VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN PERSEKUTUAN**  **KERTAS PENERANGAN**  ***(INFORMATION SHEET)***  C:\Users\norsu\Desktop\jata negara.png | | |
| **KOD DAN NAMA PROGRAM NOSS** | IT-010-3:2016 PEMBANGUNAN APLIKASI | |
| **TAHAP NOSS** | 3 | |
| **KOD, NAMA CU DAN WA NOSS** | CU01/WA3 – IMPLEMENT APPLICATION PROTOTYPE MOCK UP FLOW | |
| **NAMA PROGRAM KV** | SISTEM PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB | |
| **KOD DAN NAMA KURSUS KSKV** | KPD 1043 PRINCIPLES AND PRACTICES IN DATABASE | |
| **NO DAN TAJUK STANDARD KANDUNGAN KSKV** | K3 CREATE AND MANAGE TABLE USING DATA  DEFINITION LANGUAGE | |
| **NO. KOD NOSS** | IT-010-3:2016-C01 / P(15/26) | Muka Surat : 1 Drp : 15 |
| **NO. KOD KSKV** | KPD1043/KP(15/26) |

**TAJUK/TITLE :**

PROSEDUR MEMBINA PANGKALAN DATA MENGGUNAKAN STRUCTURE QUERY LANGUAGE (SQL)

**TUJUAN/PURPOSE :**

Di akhir pembelajaran, pelatih-pelatih mesti boleh:-

1. Terangkan prosedur membina jadual dalam pangkalan data

2. Kenal pasti sintaks arahan Structured Query Language (SQL) untuk membina

pangkalan data

**PENERANGAN**/*INFORMATION :*

**1. PROSEDUR MEMBINA JADUAL ATAU PANGKALAN DATA**

* Terdapat beberapa syarat atau perkara penting yang perlu diambil berat oleh pembangun pangkalan data. Hal ini bertujuan untuk mengelakkan ralat semasa proses membangun pangkalan data.
* Antara perkara yang perlu dititik beratkan adalah :

1. **Penamaan (*naming convention*)**

* Mempunyai *naming convention* yang konsisten di seluruh pangkalan data bermakna pembangun pangkalan data (DBA) tidak perlu untuk menghabiskan masa untuk menyemak kembali nama-nama jadual dan lajur.
* Terdapat beberapa perkara yang perlu diambil berat. Antaranya :

1. Elakkan daripada menggunakan *quotes* . Sebagai contoh : “NamaPertama” atau “SemuaPekerja”.
2. Pemboleh ubah/pengecam perlu ditulis dalam *lowercase* . Hal ini adalah untuk mengelakkan daripada berlaku kesilapan ketika proses menulis arahan dalam jadual.
3. Ruang kosong tidak dibenarkan untuk digunakan dalam pengecam/ pembolehubah
4. Tidak boleh ada simbol
5. Aksara pertama mestilah huruf atau ‘\_’
6. **Field / medan**

* Dalam jadual pangkalan data, *field* adalah struktur data.
* *Fields* disusun ke dalam rekod, yang mengandungi segala maklumat dalam jadual yang relevan kepada entiti tertentu.
* Sebagai contoh, dalam maklumat hubungan jadual pelanggan, nombor telefon akan menjadi medan berturut-turut yang juga akan mengandungi medan lain seperti alamat jalan dan bandar.

1. **Record / row / baris**

* Arahan SELECT mengunjurkan lajur *kodkursus* dan *namakursus* daripada jadual *kursus*

SELECT kodkursus,namakursus

FROM kursus

Diambil dari lajur dalam jadual

1. **Column / lajur**

* *Column* adalah satu set nilai data jenis mudah tertentu, bagi setiap baris dalam sesebuah jadual.
* Lajur menyediakan struktur data mengikut baris yang terdapat dalam jadual.
* Dalam istilah *relational database*, *column* juga turut dikenali sebagai atribut.
* Berikut adalah contoh yang menggambarkan kedudukan baris dan lajur dalam jadual di pangkalan data :

Jadual 1: Gambaran kedudukan baris dan lajur jadual

|  |  |  |
| --- | --- | --- |
|  | Lajur 1 | Lajur 2 |
| Baris 1 | Baris 1, lajur 1 | Baris 1, Lajur 2 |
| Baris 2 | Baris 2, Lajur 1 | Baris 2, Lajur 2 |
| Baris 3 | Baris 3, Lajur 1 | Baris 3, Lajur 2 |

1. **Primary Key**

* *Primary Key* adalah unik bagi mengenal pasti setiap rekod dalam jadual pangkalan data. Perhatikan pada jadual di bawah:

|  |  |
| --- | --- |
| kod\_mtp | nama\_matapelajaran |
| TI10188 | Paket pemograman 1 |
| TI66779 | Algo & Struktur Data 1 |
| TI45612 | Kalkulus |

Jadual 2: Contoh primary key

Dari contoh diatas dapat dilihat bahawa setiap data dapat dibezakan berdasarkan kolum kod\_mtp, ini berarti nilai pada kod\_mtk tidak boleh sama dengan yang lain (bersifat unik).

**Peraturan dalam primary key**

* Primary Key juga tidak boleh mengandungi nilai NULL.
* Setiap jadual hanya boleh mempunyai satu kunci utama.
* **Unik** dan **tidak boleh ganda** atau **sama**
* Berikut adalah contoh arahan yang digunakan dalam SQL bagi menggambarkan penggunaan *primary key :*

CREATE TABLE PELAJAR (  
    idpelajar int NOT NULL,  
    namapelajar varchar(265) NOT NULL,  
    kursus varchar(265),  
    umur int,  
    PRIMARY KEY (idpelajar) );

Rajah 1: Sintaks *“create table”* pelajar

1. **Auto\_increment**

* Auto\_increment digunakan bagi membolehkan nombor unik dijana secara automatik apabila rekod baru dimasukkan ke dalam jadual.
* Kebiasaannya, medan *primary key* adalah medan yang digunakan secara automatik setiap kali rekod baru dimasukkan.

1. **NULL value**

* NULL adalah penanda khas yang digunakan dalam Bahasa Pertanyaan Berstruktur (SQL) untuk menunjukkan bahawa nilai data tidak wujud dalam pangkalan data.
* Manakala jika terdapat arahan NOT NULL ianya bermaksud medan tersebut sentiasa perlu mengandungi nilai.

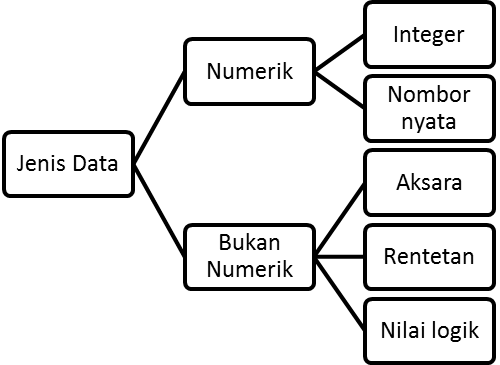
1. **Constraints**

* Kekangan (*constraints)* dalam SQL digunakan untuk menentukan kaedah-kaedah bagi data dalam jadual.
* Kekangan digunakan untuk menghadkan jenis data yang boleh digunakan dalam jadual. Ini memastikan ketepatan dan kebolehgunaan data dalam jadual.
* Berikut adalah contoh kekangan (*constraints)* yang sentiasa digunakan dalam SQL :

Jadual 3: Terminologi constraints

|  |  |
| --- | --- |
| **TERMINOLOGI** | **DEFINISI** |
| NOT NULL | Memastikan lajur tidak boleh mempunyai nilai NULL |
| UNIQUE | Memastikan semua nilai dalam lajur adalah berbeza |
| PRIMARY KEY | Gabungan NOT NULL dan UNIK bagi mengenal pasti setiap baris dalam jadual |
| FOREIGN KEY | Keunikan mengenal pasti baris / rekod dalam jadual lain |
| CHECK | Memastikan semua nilai dalam lajur memenuhi syarat tertentu |
| DEFAULT | Menetapkan nilai lalai untuk lajur apabila tiada nilai ditetapkan |
| INDEX | Gunakan untuk membuat dan mengambil data dari pangkalan data dengan cepat |

1. ***Data Types***
2. Jenis data *(Data types)* mentakrifkan jenis nilai yang boleh digunakan dalam lajur.
3. Apabila mencipta jadual, anda perlu memilih jenis data untuk setiap lajur yang akan dibentuk dalam jadual tersebut
4. Jenis data yang dipilih menentukan jenis data yang boleh disimpan dalam medan tersebut.
5. Pemilihan jenis data yang betul adalah sangat penting kerana pemilihannya memberikan kesan kepada prestasi keseluruhan pangkalan data
6. Jenis data juga menentukan jumlah ruang storan yang diperlukan.
7. Setiap lajur dalam jadual pangkalan data perlu untuk mempunyai nama dan jenis data yang tersendiri.



Rajah 2 : Jenis – jenis data

**a) Jenis Numerik :**

Semua jenis nombor

• Data yang boleh digunakan untuk pengiraan

• Terbahagi kepada

1. Interger

Semua nombor positif dan negatif

Contoh: 0, +1, -10

1. Nombor nyata (real number)

Nombor titik perpuluhan

Contoh: 1.0 cm, 234.55kg, RM20.30, 36.7C

**b) Jenis String:**

* Jenis data yang melibatkan karakter atau aksara

**c) Jenis Masa & Tarikh:**

* Jenis data untuk menyimpan tarikh dan masa

Jadual berikut menyenaraikan jenis data umum yang digunakan dalam SQL:

Jadual 4:Penerangan jenis data

|  |  |
| --- | --- |
| **JENIS DATA** | **PENERANGAN** |
| CHAR | String yang **tetap** antara 0 – 265 patah perkataan. Char digunakan untuk lajur yang mengandungi data yang tetap dan seragam. |
| VARCHAR | String yang **boleh diubah** antara 0 – 265 patah perkataan. VARCHAR digunakan dalam lajur yang mengandungi data yang tidak seragam. |
| TINYTEXT | String dengan panjang maksimum 265 patah perkataan |
| TEXT | String dengan maksimum 65,535 patah perkataan |
| INT | Nombor dalam julat -2,147,483,648 hingga 2,147,483,648 atau  Positif 0 hingga 4,294,967,275 |
| FLOAT | Nombor kecil yang mempunyai titik perpuluhan |
| DOUBLE | Nombor besar yang mempunyai titik perpuluhan |
| DATE | Tarikh dengan format YYYY-MM-DD |
| DATETIME | Tarikh dan waktu dengan format YYYY-MM-DD/HH:MM:SS |
| TIME | Waktu dengan format HH:MM:SS |

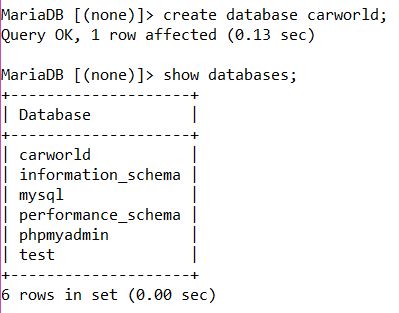
**2. SINTAKS ARAHAN STRUCTURED QUERY LANGUAGE**

**2.1 Penyataan CREATE DATABASE**

* Digunakan untuk membina pangkalan data.
* Berikut adalah sintaks bagi arahan ini :

**CREATE DATABASE** nama\_pangkalan\_data;

Penyataan 1: Arahan bina pangkalan data

* Contoh :

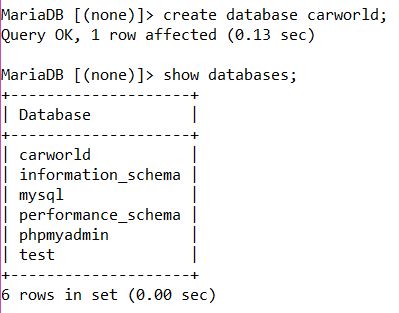
Rajah 3: Sintaks *create database* carworld

**2.2 Penyataan SHOW DATABASES**

* Digunakan untuk memaparkan pangkalan data yang telah dibina.
* Sintaks:

**SHOW DATABASES;**

Penyataan 2: Arahan papar pangkalan data

* Contoh :

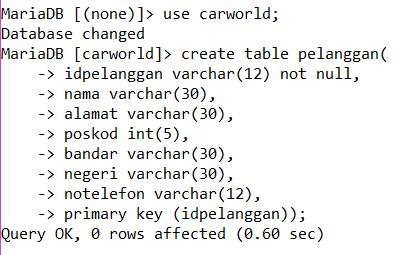
Rajah 4: Sintaks papar pangkalan data

**2.3 Penyataan USE**

* Digunakan untuk memilih pangkalan data yang hendak digunakan.
* Sintaks:

**USE** nama\_pangkalan\_data

Penyataan 3: Arahan guna pangkalan data

* Contoh :

Rajah 5: Sintaks guna pangkalan data

**2.4 Penyataan CREATE TABLE**

Penyataan CREATE TABLE digunakan untuk :

* Membina jadual
* Menakrifkan jadual
* Menakrifkan lajur / column

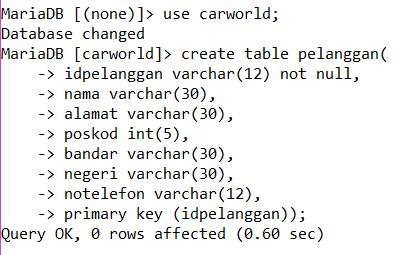
Berikut adalah Sintaksbagi arahan ini :

**CREATE TABLE** nama\_jadual

(takrifan\_lajur) ;

Penyataan 4: Arahan bina jadual

* Contoh :



Rajah 6: Sintaks bina jadual

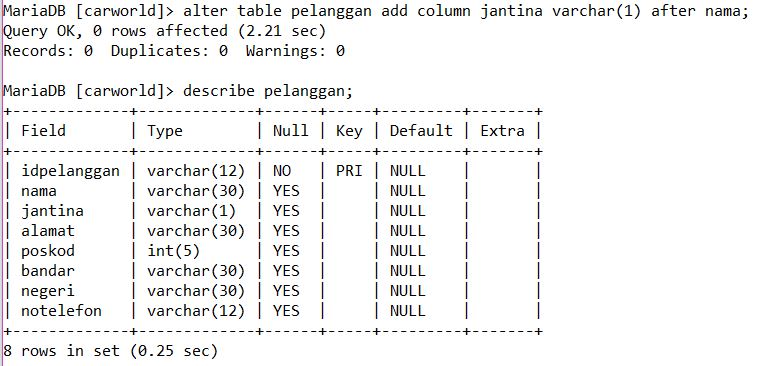
**2.5 Penyataan DESCRIBE**

* Penyataan DESCRIBE digunakan untuk memaparkan medan-medan yang telah didefinisikan dalam jadual.
* Sintaks:

**DESCRIBE** nama\_jadual;

Penyataan 5: Arahan papar jadual

* Contoh :



Rajah 7: Paparan jadual pelanggan yang telah dibina

**2.6 Penyataan ALTER**

Digunakan untuk menambah, menghapuskan atau mengubahsuai lajur (column) dalam jadual yang telah dibina.

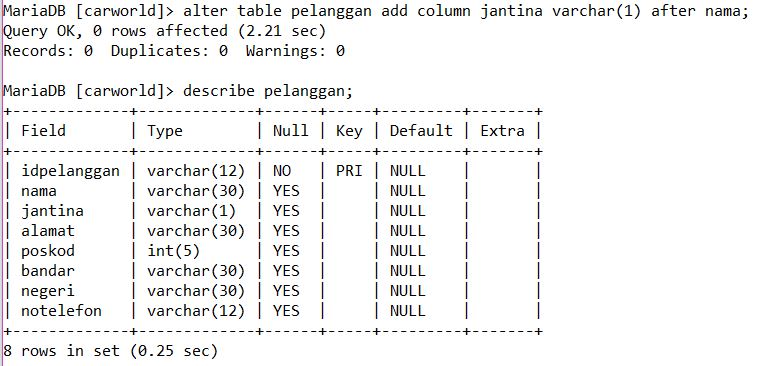
1. Penyataan ALTER TABLE – ADD COLUMN

* Digunakan untuk menambah lajur dalam jadual yang sedia ada.
* Berikut adalah sintaks bagi arahan ini :

**ALTER TABLE** nama\_jadual **ADD** nama\_lajur jenis\_data ;

Penyataan 6: Arahan tambah lajur dalam jadual

* Contoh :



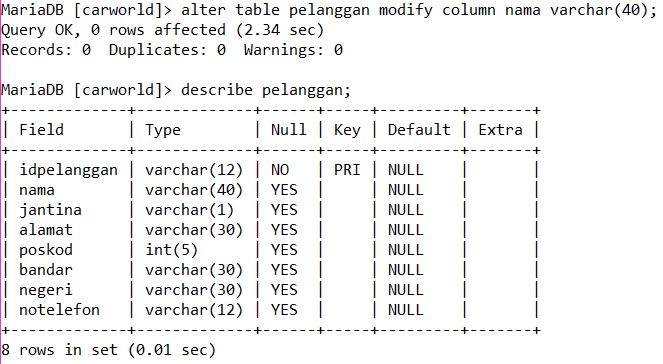
Rajah 8: Paparan tambah baris jantina pada jadual pelanggan

1. Penyataan ALTER TABLE – MODIFY COLUMN

* Digunakan untuk mengubahsuai jenis dan saiz data bagi lajur(column) dalam jadual yang sedia ada.
* Sintaks :

**ALTER TABLE** nama\_jadual **MODIFY** nama\_lajur definisi\_baharu;

Penyataan 7: Arahan ubah j jenis dan saiz data

* Contoh :

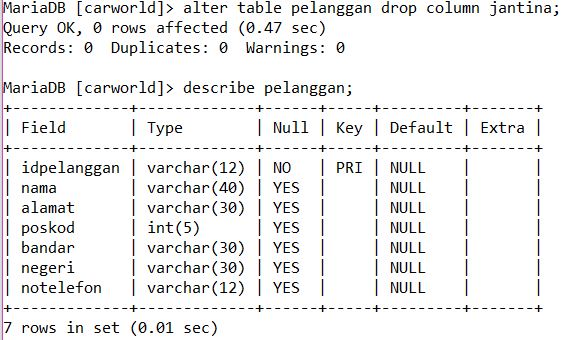
Rajah 9: Paparan tambah baris jantina pada jadual pelanggan

1. Penyataan ALTER TABLE – DROP COLUMN

* Digunakan untuk menghapuskan lajur dalam jadual.
* Sintaks :

**ALTER TABLE** nama\_jadual **DROP** nama\_lajur;

* Contoh :



**2.7 Penyataan DROP**

* Digunakan untuk menyingkir atau menghapuskan jadual dan pangkalan data yang sedia ada.
* Penyataan DROP TABLE digunakan untuk menghapuskan jadual yang sedia ada.
* Berikut adalah sintaks bagi arahan ini :

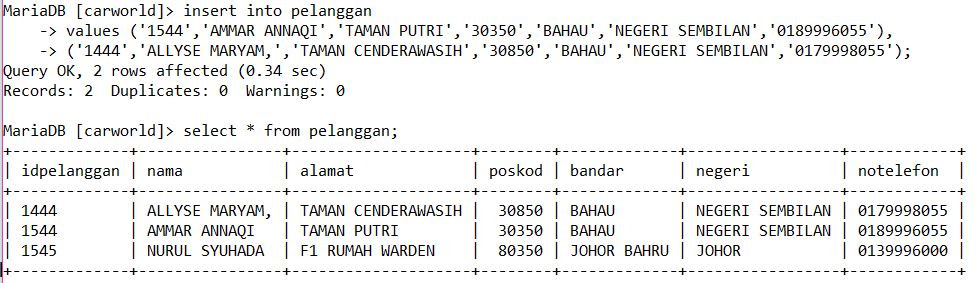
**DROP TABLE** nama\_jadual;

**2.8 Penyataan INSERT**

* Menyisipkan satu baris baru ke dalam jadual sedia ada melalui ungkapan.

**INSERT INTO** nama jadual ( nama lajur1, lajur 2, lajur 3)

**VALUES** (‘data1’ , ‘data 2’ , ’data 3’)

* Contoh :

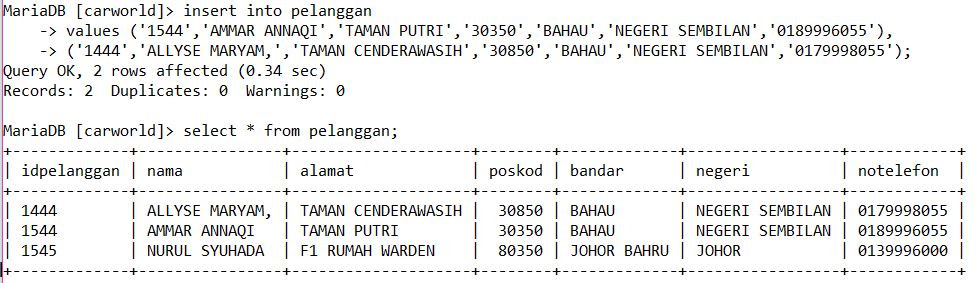
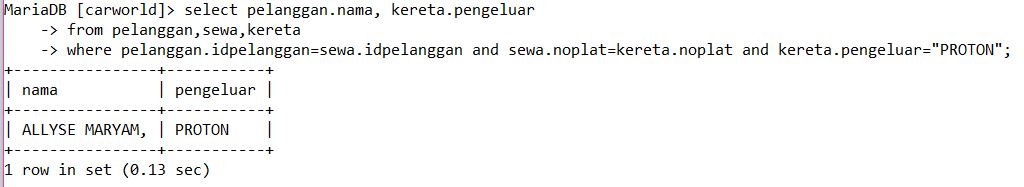
**2. 9 Penyataan SELECT**

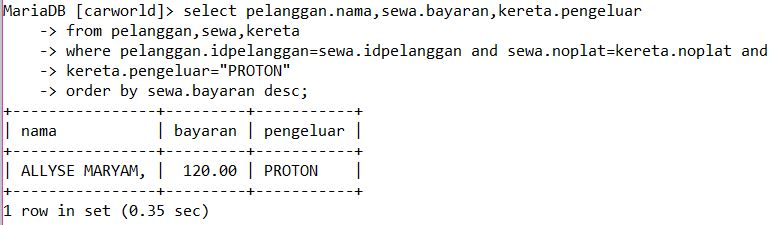
* Digunakan untuk mendapatkan data atau memaparkan data daripada satu jadual atau gabungan jadual.

**SELECT**  nama lajur

**FROM** nama jadual

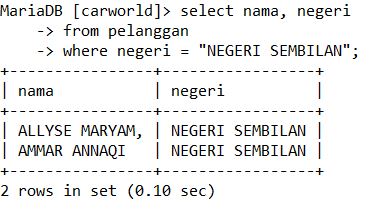
**WHERE** tertakluk kepada syarat yang telah ditentukan

* Contoh :
* Contoh untuk senaraikan nama pelanggan yang menyewa kereta proton.
* Contoh untuk senaraikan nama, bayaran dan pengeluar bagi pelanggan yang menyewa kereta proton. Bayaran pelanggan ini disusun secara menurun.



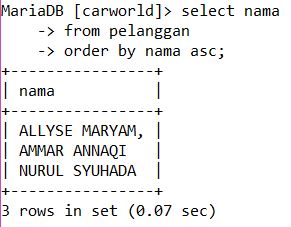
**2.10 Penyataan WHERE**

* Menentukan rekod yang ingin dipaparkan. Semua atribut dalam rekod akan dipaparkan jika tiada perkataan **WHERE.**
* Contoh arahan jika ingin memaparkan pelanggan dari “ negeri sembilan “ sahaja.

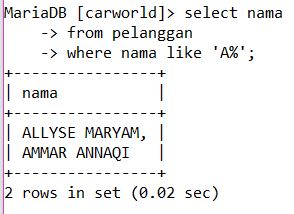


**2.11 Penyataan ORDER BY**

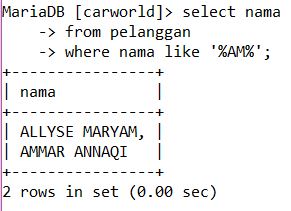
* Mengisih rekod mengikut susunan menaik.
* *desc* digunakan untuk mengisih rekod mengikut susunan menurun
* Contoh arahan ingin memaparkan senarai nama pelanggan secara menaik menggunakan arahan *order by*



* Contoh arahan ingin memaparkan senarai nama pelanggan yang bermula dengan huruf **A** menggunakan arahan *like*



* Contoh arahan untuk memaparkan nama pelanggan yang mengandungi huruf am



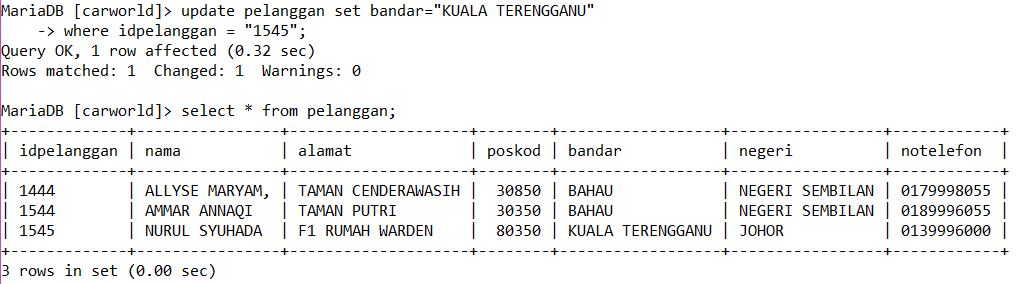
**2.12 Penyataan UPDATE**

* Untuk mengemaskini data dalam jadual

**UPDATE**  nama jadual

**SET**  namalajur1 = nilai data1

**WHERE** tertakluk kepada syarat yang telah ditentukan

* Contoh :

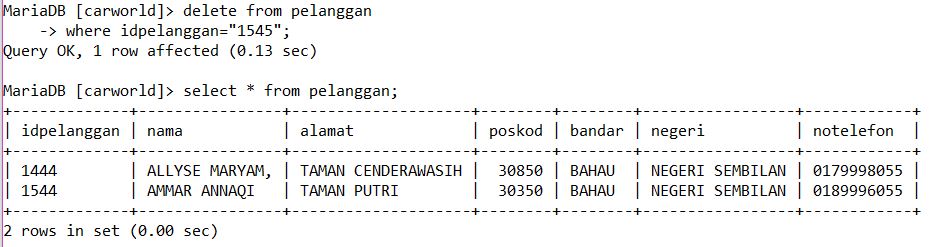
**2.13 Penyataan DELETE**

* Untuk menghapuskan data daripada jadual.

**DELETE FROM**  nama jadual

**WHERE** tertakluk kepada syarat yang telah ditentukan

* Contoh :



**2.14 Penyataan Transmission Control Command**

**a) Penyataan COMMIT**

* Arahan *COMMIT* digunakan untuk menyimpan perubahan yang digunakan dalam pangkalan data.
* Berikut adalah contoh penggunaan arahan *commit*

**DELETE FROM** nama jadual

**WHERE** namalajur = syarat;

**COMMIT**;

**b) Penyataan ROLLBACK dalam TCC**

* Arahan *rollback* mengembalikansegala urusan transaksi yang digunakan untuk membatalkan transaksi yang belum lagi disimpan ke dalam pangkalan data.
* Berikut adalah contoh penggunaan arahan *rollback*

**DELETE FROM** nama jadual

**WHERE** namalajur = syarat;

**ROLLBACK;**

**c) Penyataan SAVEPOINT dalam TCC**

* *SAVEPOINT* adalah titik dalam proses transaksi apabila pembangun boleh mengembalikan transaksi ke titik tertentu tanpa *ROLLBACK* secara keseluruhan.
* Berikut adalah contoh penggunaan arahan *savepoint.*

**SAVEPOINT SAVEPOINT\_NAME;**

**SOALAN**

1**.** Nyatakan tujuan penyataan CREATE.

2. Jawab soalan di bawah dengan merujuk jadual Pesakit dalam pangkalan data Hospital\_Bahagia.

|  |  |  |
| --- | --- | --- |
| **Nama lajur (*column*)** | **Jenis data ( saiz data )** | ***Constraint*** |
| No\_KP | Int(14) | Auto Increment  Primary Key |
| Nama\_Pesakit | varchar(50) | Not Null |
| Wad\_Pesakit | varchar(20) | Not Null |
| Nama\_Penyakit | varchar(100) |  |

Jadual Pesakit

1. Tuliskan arahan SQL untuk menghapuskan jadual Pesakit.
2. Tuliskan arahan SQL untuk menghapuskan pangkalan data tersebut.

RUJUKAN/REFERENCE :

1. Abdullah Embong, 2000, sistem pangkalan data konsep asas, rekabentuk da pelaksanaan, tradisi ilmu sdb.bhd
2. Cosmopoint Institut of information technology , 2001, data analysis & database design, Penerbitan cosmopoint
3. Halina Mohamed Dahlan, 2011, Sistem Pangkalan Data, Penerbitan UTM Press Sdn. Bhd, (ms 35 - 48)
4. Modi Lakulu, Aslina Saad, 2007, Sistem Pangkalan Data, Universiti Pendidikan Sultan Idris
5. Thomas Connolly, Database Systems 2005, Pearson Education Limited Sdn. Bhd, (ms 34)