

**LAPORAN TUBES**  
**TUNING DATABASE SYSTEM**  
**MANAGEMENT BASIS DATA**



Oleh:

**Shella Intia**

**14116070**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI PRODUKSI, INDUSTRI, DAN INFORMASI**  
**INSTITUT TEKNOLOGI SUMATERA**

**2019**

**BAB 1**

# PENDAHULUAN

## 1.1.Latar Belakang

Dalam kehidupan sehari-hari kita, sebagai masyarakat informasi, kita selalu memproduksi dan mengonsumsi data dan informasi baik sebagai individu, sebagai lembaga, maupun sebagai pelaku bisnis. Bahkan beberapa lembaga tidak akan berfungsi bila tidak didukung oleh data dan informasi, misalnya pemerintah, bank, masmedia, dan industri. Pengelola lembaga-lembaga ini berharap mendapatkan informasi yang tepat, akurat dan pada saat yang tepat. Secara langsung basis data dan sistem basis data menjadi komponen utama dalam kehidupan masyarakat modern saat ini. Berbagai aktivitas manusia secara tidak langsung berhubungan dengan database. Sebab, database tidak lain adalah record-keeping system yang berbasis komputer.

Konsep utama dari sistem basis data adalah ide pemisahan antara Program Aplikasi dengan jumlah data sebenarnya yang disimpan. Konsep ini lebih dikenal sebagai Kebebasan Program Data atau biasa disingkat Kebebasan Data (Data Independence). Adanya kebebasan data pada sistem basis data (Database Manajement System) memberi kemungkinan untuk mengatasi masalah “pemeliharaan yang tidak produktif (unproductive maintenance)”. Pemeliharaan data misalnya, menyangkut penambahan item.

Dengan adanya sistem basis data, memudahkan dalam mencari informasi (kumpulan data yang telah mengalami proses). Data-data tersebut diatur/diorganisir secara tepat, sehingga dapat dengan cepat memperoleh informasi dari susunan data yang diinginkan.

## 1.2.Tujuan

Tujuan dibuatnya tugas besar ini adalah:

1. Mengimplementasikan ilmu yang sudah di dapat selama perkuliahan.
2. Memenuhi tugas besar pada matakuliah Management Basis Data (MBD).
3. Mengasah kemampuan mengenai konsep dan juga cara melakukan tuning.

### 1.3.Manfaat

Pemanfaatan perangkat komputer dalam sebuah organisasi/perusahaan hampir selalu berhubungan dengan basis data, dan biasanya digunakan untuk menjalankan fungsi pengelolaan sistem informasi. Dengan adanya basis data sebagai salah satu komponen utama dalam setiap sistem informasi, maka diharapkan dapat meningkatkan *efisiensi*, daya saing, *keakuratan*, *kecepatan* operasional organisasi/perusahaan terhadap sejumlah data yang telah diatur/diorganisir tersebut.

Jadi secara umum, basis data ini dapat dimanfaatkan untuk membuat/menjalankan suatu sistem informasi, memudahkan dalam proses pencarian dalam kumpulan data, dll.

## **BAB II**

### **DASAR TEORI**

#### **2.1. Pengertian Basis Data**

Basis data atau dalam Bahasa Inggris disebut *database*, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Istilah basis data mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai sistem manajemen basis data (*database management system/DBMS*). Jika konteksnya sudah jelas, banyak administrator dan programmer menggunakan istilah basis data untuk kedua arti tersebut.

## 2.2.Database Management System (DBMS)

Menurut Connolly & Begg (2010, p66), DBMS adalah suatu system *software* yang memberikan kebebasan pada *user* untuk mendefinisikan, membuat, memelihara dan mengontrol akses ke *database*. Sedangkan menurut Satzinger, et al (2005, p398) DBMS merupakan suatu sistem piranti lunak yang melakukan pengelolaan dan kontrol tiap-tiap akses yang dilakukan pada *database*. Menurut Connolly dan Begg, (Connolly & Begg, 2010, p. 127), komponen dari sebuah DBMS adalah sebagai berikut :

- *Query Processor*

Merupakan komponen DBMS yang utama yang mengubah *query* ke dalam seperangkat instruksi tingkat rendah langsung ke *database manager*.

- *Database Manager*

*Database Manager* menghubungkan program aplikasi *user-submitted* dan *query*. *Database Manager* menerima *query* dan memeriksa skema *eksternal* dan konseptual untuk menentukan *record* konseptual apa yang diperlukan untuk memuaskan permintaan.

- *File Manager*

*File Manager* memanipulasi penyimpanan *file* dan mengatur penempatan ruang penyimpanan dalam *disk*. Komponen ini mendirikan dan memelihara daftar struktur dan indeks yang didefinisikan dalam skema *internal*.

- *DML Preprocessor*

Modul ini mengubah pernyataan DML yang tertanam dalam program aplikasi ke dalam panggilan fungsi *standard* dalam *host language*. Komponen ini harus berinteraksi dengan *query processor* untuk membuat kode yang sesuai.

- *DDL Compiler*

Modul ini mengubah pernyataan DDL ke dalam seperangkat tabel berisi *metadata*. Tabel ini kemudian disimpan dalam katalog sistem sementara itu informasi kendali disimpan dalam *handler file* data.

- *Catalog Manager*

Mengatur akses dan memelihara katalog sistem. Katalog sistem diakses oleh sebagian besar komponen DBMS.

### **2.2.1. Keuntungan dan kerugian *Database Management System***

Menurut Connolly (2010, p77) manajemen basis data memiliki banyak potensi yang menjanjikan, tetapi juga memiliki beberapa kelemahan. Berikut adalah beberapa kelemahan dan kelebihan dari manajemen basis data.

Kelebihan yang dimiliki oleh Sistem Manajemen Basis Data:

- *Control of Data Redudancy*  
*File-based system* banyak memakan ruang dengan menyimpan informasi sama di dalam satu berkas atau lebih.
- *Data Consistency*  
Dengan menghilangkan atau mengendalikan redudansi, akan mengurangi resiko adanya ketidakstabilan data.
- Banyak informasi dari jumlah data yang sama  
Dengan mengintegrasikan data operasional, maka memungkinkan organisasi untuk mendapat informasi dari data yang sama.

Kekurangan yang dimiliki oleh Sistem Manajemen Basis Data.

- *Complexity*  
Ketentuan dari fungsi yang diharapkan dari DBMS yang bagus membuatnya menjadi sistem yang rumit.
- *Size*  
Kerumitan dan banyaknya hal yang bisa dikerjakan membuat DBMS menjadi *software* yang besar.
- *Cost of DBMS*  
Harga untuk DBMS yang besar cukup mahal, berkisar antara \$100,000 sampai dengan \$1,000,000.
- *Performance*

DBMS mencakup banyak aplikasi, sehingga beberapa aplikasi mungkin akan berjalan lebih lambat.

### 2.3. Spesifikasi Alat

#### 1. Java

Java adalah Bahasa pemrograman tingkat tinggi yang berorientasi objek dan program java tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Pemrograman java banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas Java, yang disebut dengan *Java Application Programming Interface* (API).

#### 2. phpMyAdmin



Gambar 2. Logo phpMyAdmin

Source: Wikipedia.org

phpMyAdmin adalah perangkat lunak bebas yang ditulis dalam Bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui *World Wide Web* (www). phpMyAdmin mendukung administrasi MySQL, diantaranya (mengelola basisdata, table-tabel, relasi, indeks, pengguna, perijinan, dan lain-lain).

### 3. MySQL



Gambar 1. Logo MySQL

Source: acamedia.edu

Menurut Welling, Luke dan Thompson, Laura. (2003:4) *MySQL* bersifat cepat, kuat dan juga merupakan *Relational Database Management System* (RDBMS). Pengguna dapat menyimpan, mencari, dan menyortir data secara efisien. *MySQL* server mengontrol akses ke data pengguna untuk menjamin penggunaan secara bersamaan antar pengguna, juga menyediakan akses cepat ke *database* dan menjamin bahwa hanya pengguna yang berwenang saja yang mendapatkan hak akses kedalam *database*. Beberapa keuntungan penggunaan *MySQL* diantaranya yaitu :

- Performa yang tinggi Performa yang sangat tinggi memungkinkan pengguna untuk bekerja secara cepat juga.
- Biaya rendah Tidak dikenakan biaya apapun, lisensi bersifat *open surce*.
- Mudah untuk dipelajari dan digunakan Saat ini banyak *database* menggunakan SQL, namun SQL tetap mudah untuk digunakan pada produk yang serupa.
- Bersifat *portable MySQL* bisa digunakan di beberapa sistem operasi, seperti Linux dan Windows.
- Ketersediaan *source code* Sama seperti PHP, pengguna juga bisa memodifikasi *source code MySQL*.

#### 2.4.Alat yang Digunakan

Pada tugas besar ini penulis menggunakan laptop ASUS *Republic Of Gamers* (ROG) dengan spesifikasi sebagai berikut:



1. Processor : Intel® Core™ i7-6700HQ CPU @ 2.60GHz 2.59GHz
2. RAM : 8.00 GB
3. Storage : 250 GB SSD + 1 TB HDD
4. OS : Windows 10 Home Single Language

## **BAB III**

### **PERCOBAAN**

#### **3.1. Proses Generate Data**

Pada proses generate data menggunakan tableGen.java dan dilakukan dengan perintah sebagai berikut:

- javac tableGen.java
- java tableGen

#### **3.2. Proses Perubahan Nilai**

Proses perubahan nilai pada setiap data dengan nilai yang telah ditentukan

advisor = 100, student = 100, section = 200,takes = 200
advisor = 200, student = 200, section = 400,takes = 400
advisor = 500, student = 500, section = 1000,takes = 1000
advisor = 700, student = 700, section = 20000,takes = 20000
advisor = 1000, student = 1000, section = 100000,takes = 1000000
advisor = 1800, student = 1800, section = 180000,takes = 1800000
advisor = 10000, student = 10000, section = 30000000,takes = 30000000

#### **3.3. Proses Tuning**

1. Proses tuning dengan mengubah konfigurasi pada my.in.

Nilai yang diubah yaitu:

- innodb\_buffer\_pool\_size dari 16M menjadi 2700M
- innodb\_additional\_mem\_pool\_size dari 2M menjadi 20M
- sort\_buffer\_size dari 20M menjadi 512K

2. Tuning dengan menambah index pada kolom table :

- Kolom name pada tabel student
- Kolom tot\_cred pada tabel student
- Kolom grade pada tabel takes

Kolom time\_slot\_id pada tabel section

### 3.4.Percobaan

Table 1. SELECT \* FROM student;

No.	Jumlah Data	Sebelum (second)	Config (second)	Index (second)
1.	advisor = 100, student = 100, section = 200, takes = 200	0,00	0,00	0,00
2.	advisor = 200, student = 200, section = 400, takes = 400	0,00	0,00	0,00
3.	advisor = 500, student = 500, section = 1000, takes = 1000	0,04	0,01	0,00
4.	advisor = 700, student = 700, section = 20000, takes = 20000	0,00	0,00	0,00
5.	advisor = 1000, student = 1000, section = 100000, takes = 1000000	0,00	0,00	0,00
6.	advisor = 1800, student = 1800, section = 180000, takes = 1800000	0,06	0,02	0,00
7.	advisor = 10000, student = 10000, section = 30000000, takes = 30000000	0,00	0,00	0,00

Pada table 1 diatas didapatkan pengujian 1, 2, 4, 5, dan 7 sebelum tuning, tuning dengan config, dan tuning dengan index nilainya sama yaitu 0,00 second. Pada percobaan 3 memiliki nilai sebelum tuning 0,04 second, tuning dengan config 0,01 second, dan tuning dengan index 0,00 second. Pada percobaan 6 memiliki nilai sebelum tuning 0,06 second, tuning dengan config 0,02 second, dan tuning dengan index 0,00 second.

Tabel 2. SELECT \* FROM student WHERE tot\_cred > 30;

No.	Jumlah Data	Sebelum (second)	Config (second)	Index (second)
1.	advisor = 100, student = 100, section = 200, takes = 200	0,00	0,00	0,00
2.	advisor = 200, student = 200, section = 400, takes = 400	0,00	0,00	0,00
3.	advisor = 500, student = 500, section = 1000, takes = 1000	0,00	0,00	0,00
4.	advisor = 700, student = 700, section = 20000, takes = 20000	0,00	0,00	0,00
5.	advisor = 1000, student = 1000, section = 100000, takes = 1000000	0,00	0,00	0,00
6.	advisor = 1800, student = 1800, section = 180000, takes = 1800000	0,00	0,00	0,00

7.	advisor = 10000, student = 10000, section = 300000000, takes = 300000000	0,00	0,00	0,00
----	--	------	------	------

Pada table 2 diatas didapatkan pengujian 1 sampai dengan pengujian 7 sebelum tuning, tuning dengan config, dan tuning dengan index memiliki nilai yang sama yaitu 0,00 second.

Tabel 3. SELECT name, dept\_name FROM student WHERE tot\_cred > 30;

No.	Jumlah Data	Sebelum (second)	Config (second)	Index (second)
1.	advisor = 100, student = 100, section = 200, takes = 200	0,00	0,00	0,00
2.	advisor = 200, student = 200, section = 400, takes = 400	0,00	0,00	0,00
3.	advisor = 500, student = 500, section = 1000, takes = 1000	0,00	0,00	0,00
4.	advisor = 700, student = 700, section = 20000, takes = 20000	0,00	0,00	0,00

5.	advisor = 1000, student = 1000, section = 100000, takes = 1000000	0,00	0,00	0,00
6.	advisor = 1800, student = 1800, section = 180000, takes = 1800000	0,00	0,00	0,00
7.	advisor = 10000, student = 10000, section = 30000000, takes = 30000000	0,00	0,00	0,00

Pada table 3 diatas didapatkan pengujian 1 sampai dengan pengujian 7 sebelum tuning, tuning dengan config, dan tuning dengan index memiliki nilai yang sama yaitu 0,00 second.

Tabel 4. SELECT \* FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course\_id = section.course\_id;

No.	Jumlah Data	Sebelum (second)	Config (second)	Index (second)
1.	advisor = 100, student = 100, section = 200, takes = 200	0,00	0,00	0,00
2.	advisor = 200, student = 200, section = 400, takes = 400	0,01	0,01	0,01

3.	advisor = 500, student = 500, section = 1000, takes = 1000	0,14	0,09	0,04
4.	advisor = 700, student = 700, section = 20000, takes = 20000	1,06	0,12	0,07
5.	advisor = 1000, student = 1000, section = 100000, takes = 1000000	0,04	0,00	0,00
6.	advisor = 1800, student = 1800, section = 180000, takes = 1800000	0,08	0,03	0,00
7.	advisor = 10000, student = 10000, section = 30000000, takes = 30000000	0,07	0,05	0,05

Pada table 4 diatas didapatkan pengujian 1 sampai dengan pengujian 7 hasilnya cukup bervariasi pada percobaan 1 sebelum tuning, tuning dengan config, dan tuning dengan index sama-sama 0,00 second. pada percobaan 2 sebelum tuning, tuning dengan config, dan tuning dengan index sama-sama 0,01 second. pada percobaan 3 sebelum tuning 0,14 second, tuning dengan config 0,09 second, dan tuning dengan index 0,04 second. Begitupun pada percobaan 4 sampai 7 memiliki nilai cukup beragam.

Tabel 5. SELECT student.`name`,student.dept\_name,takes.sec\_id AS  
pengambilan,takes.semester,section.room\_number,section.building,course.course\_id,course.dept  
\_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course\_id =  
section.course\_id JOIN course ON section.course\_id = course.course\_id;

No.	Jumlah Data	Sebelum (second)	Config (second)	Index (second)
1.	advisor = 100, student = 100, section = 200, takes = 200	0,06	0,01	0,01
2.	advisor = 200, student = 200, section = 400, takes = 400	0,04	0,01	0,01
3.	advisor = 500, student = 500, section = 1000, takes = 1000	0,06	0,03	0,01
4.	advisor = 700, student = 700, section = 20000, takes = 20000	0,02	0,00	0,00
5.	advisor = 1000, student = 1000, section = 100000, takes = 1000000	0,02	0,02	0,02
6.	advisor = 1800, student = 1800, section =	0,02	0,01	0,00



	180000, takes = 1800000			
7.	advisor = 10000, student = 10000, section = 30000000, takes = 30000000	0,07	0,03	0,01

Pada table 5 diatas didapatkan pengujian 1 sampai dengan pengujian 7 hasilnya cukup bervariasi pada percobaan 1 sebelum tuning 0,06 second, tuning dengan config, dan tuning dengan index sama-sama 0,00 second. pada percobaan 2 sebelum tuning 0,04 second, tuning dengan config, dan tuning dengan index sama-sama 0,01 second. pada percobaan 3 sebelum tuning 0,06 second, tuning dengan config 0,03 second, dan tuning dengan index 0,01 second. Begitupun pada percobaan 4 sampai 7 memiliki nilai yang cukup beragam.

## **BAB IV**

### **PENUTUP**

#### **4.1.Kesimpulan**

Dari data yang terdapat diatas didapatkan hasil pada query 1 sampai dengan query 3 mendapatkan respon time yang cukup cepat yaitu 0,00 second dan yang paling lama adalah 0,18 second karena pada query ini cukup ringan dan tidak terlalu membebani kinerja DBMS dan juga dikarenakan hanya ada operasi SELECT yang berguna untuk menampilkan data. Untuk query 4 dan 5 mendapatkan respon time yang lebih lambat, hingga terlama 4,25 second, hal itu dikarenakan pada query tersebut menggunakan operasi JOIN yang sangat membebani kinerja pada DBMS.

Untuk hasil tuning dan index lebih cepat dibandingkan dengan config. Untuk data yang sangat besar, index dapat sangat membantu kinerja DBMS karena data disimpan dalam bentuk pohon sehingga dapat membantu dalam pembacaan data.

## LAMPIRAN

```
Command Prompt - mysql -u root
```

48355	rahmat	HH	70
49052	Ahmad	RQ	7
49925	Adri	DE	115
51428	Josu	HH	56
51621	Adri	WW	24
52099	yuyun	DE	56
52460	yuyun	GO	111
52578	Yohan	ED	73
53188	yuyun	ED	29
53950	Ande	EL	71
5629	Yohan	RQ	62
595	yuyun	DF	53
62005	Budi	RR	35
62058	rahmat	HH	55
64668	Yohan	GO	34
65050	Adri	RR	128
65370	rahmat	RQ	129
67104	Budi	GO	104
67488	Kiki	WW	4
67834	Ahmad	GO	95
68831	Budi	RQ	74
71036	Ahmad	DE	23
71203	Yohan	EL	51
71851	rahmat	DF	42
72082	Budi	WW	4
72385	Adri	RQ	80
73062	Adri	DK	101
73501	Johan	RQ	11
73866	rahmat	HH	101
74788	Yohan	GO	26
75581	Johan	DE	60
76193	Yohan	ED	63
76828	Kiki	DK	93
80110	Josu	WW	4
80408	Adri	DK	5
80796	Adri	GO	0
81731	yuyun	ED	111
82109	Ande	DF	56
82897	Kiki	RQ	1
83923	rahmat	HH	23
84432	rahmat	DE	34
88784	Johan	WW	82
90178	yuyun	EL	37
90192	Ande	WW	67
90517	Budi	GO	127
90774	Yohan	HH	24
90788	Ahmad	DE	88
91029	Ahmad	RQ	128
95165	Adri	DF	116
96384	yuyun	GO	5
96999	Kiki	RR	56
97383	Yohan	EL	58
97764	Adri	ED	48
98308	Budi	RR	129
99273	Yohan	RR	12
99508	Yohan	DF	41

100 rows in set (0.00 sec)

Command Prompt - mysql -u root

23847	Ande	RQ	53
28479	Johan	RR	71
28493	Johan	EL	85
28551	Josu	RR	78
30481	Ahmad	DK	100
31158	Adri	WW	69
31311	Adri	RQ	74
32244	Budi	ED	68
32512	Ahmad	DE	121
32569	Ande	HH	96
33209	Josu	WW	97
35587	Ahmad	DK	119
36086	rahmat	RQ	66
37239	Johan	EL	48
37835	Johan	ED	95
40393	yuyun	WW	59
43173	Kiki	DK	123
43692	yuyun	DK	38
45627	Budi	EL	74
46791	Adri	WW	37
47798	rahmat	RQ	86
48355	rahmat	HH	70
49925	Adri	DE	115
51428	Josu	HH	56
52099	yuyun	DE	56
52460	yuyun	GO	111
52578	Yohan	ED	73
53950	Ande	EL	71
5629	Yohan	RQ	62
595	yuyun	DF	53
62005	Budi	RR	35
62058	rahmat	HH	55
64668	Yohan	GO	34
65050	Adri	RR	128
65370	rahmat	RQ	129
67104	Budi	GO	104
67834	Ahmad	GO	95
68831	Budi	RQ	74
71203	Yohan	EL	51
71851	rahmat	DF	42
72385	Adri	RQ	80
73062	Adri	DK	101
73866	rahmat	HH	101
75581	Johan	DE	60
76193	Yohan	ED	63
76828	Kiki	DK	93
81731	yuyun	ED	111
82109	Ande	DF	56
84432	rahmat	DE	34
88784	Johan	WW	82
90178	yuyun	EL	37
90192	Ande	WW	67
90517	Budi	GO	127
90788	Ahmad	DE	88
91029	Ahmad	RQ	128
95165	Adri	DF	116
96999	Kiki	RR	56
97383	Yohan	EL	58
97764	Adri	ED	48
98308	Budi	RR	129
99508	Yohan	DF	41

77 rows in set (0.00 sec)

Command Prompt - mysql -u root

Johan	EL
Josu	RR
Ahmad	DK
Adri	WW
Adri	RQ
Budi	ED
Ahmad	DE
Ande	HH
Josu	WW
Ahmad	DK
rahmat	RQ
Johan	EL
Johan	ED
yuyun	WW
Kiki	DK
yuyun	DK
Budi	EL
Adri	WW
rahmat	RQ
rahmat	HH
Adri	DE
Josu	HH
yuyun	DE
yuyun	GO
Yohan	ED
Ande	EL
Yohan	RQ
yuyun	DF
Budi	RR
rahmat	HH
Yohan	GO
Adri	RR
rahmat	RQ
Budi	GO
Ahmad	GO
Budi	RQ
Yohan	EL
rahmat	DF
Adri	RQ
Adri	DK
rahmat	HH
Johan	DE
Yohan	ED
Kiki	DK
yuyun	ED
Ande	DF
rahmat	DE
Johan	WW
yuyun	EL
Ande	WW
Budi	GO
Ahmad	DE
Ahmad	RQ
Adri	DF
Kiki	RR
Yohan	EL
Adri	ED
Budi	RR
Yohan	DF

77 rows in set (0.00 sec)

Command Prompt - mysql -u root

82109	834	1	Fall	2007	C	82109	Ande	DF	56	834	2	Fall	2002	D	499	N
82109	834	1	Fall	2007	C	82109	Ande	DF	56	834	3	Spring	2003	D	454	A
82897	311	1	Spring	2001	C	82897	Kiki	RQ	1	311	1	Spring	2001	F	352	O
83923	582	2	Spring	2002	C+	83923	rahmat	HH	23	582	1	Spring	2005	F	352	O
83923	582	2	Spring	2002	C+	83923	rahmat	HH	23	582	2	Spring	2002	G	239	L
83923	697	1	Spring	2006	A+	83923	rahmat	HH	23	697	1	Spring	2006	D	454	E
83923	787	1	Spring	2001	C+	83923	rahmat	HH	23	787	1	Spring	2001	D	454	L
83923	787	1	Spring	2001	C+	83923	rahmat	HH	23	787	2	Spring	2004	G	239	K
88784	201	3	Spring	2001	A-	88784	Johan	WW	82	201	1	Spring	2007	F	352	J
88784	201	3	Spring	2001	A-	88784	Johan	WW	82	201	2	Fall	2001	D	583	M
88784	201	3	Spring	2001	A-	88784	Johan	WW	82	201	3	Spring	2001	D	583	A
90178	661	1	Fall	2002	A-	90178	yuyun	EL	37	661	1	Fall	2002	J	346	I
90192	578	1	Spring	2010	A	90192	Ande	WW	67	578	1	Spring	2010	J	334	D
90192	697	1	Spring	2006	A+	90192	Ande	WW	67	697	1	Spring	2006	D	454	E
90192	702	2	Spring	2007	C+	90192	Ande	WW	67	702	1	Spring	2003	A	517	A
90192	702	2	Spring	2007	C+	90192	Ande	WW	67	702	2	Spring	2007	I	104	J
90192	787	2	Spring	2004	A+	90192	Ande	WW	67	787	1	Spring	2001	D	454	L
90192	787	2	Spring	2004	A+	90192	Ande	WW	67	787	2	Spring	2004	G	239	K
90517	116	1	Spring	2010	B+	90517	Budi	GO	127	116	1	Spring	2010	I	104	L
90517	546	2	Fall	2005	C+	90517	Budi	GO	127	546	1	Spring	2002	D	454	H
90517	546	2	Fall	2005	C+	90517	Budi	GO	127	546	2	Fall	2005	I	104	D
90517	913	3	Fall	2009	C+	90517	Budi	GO	127	913	1	Spring	2003	J	334	K
90517	913	3	Fall	2009	C+	90517	Budi	GO	127	913	2	Spring	2008	J	346	A
90517	913	3	Fall	2009	C+	90517	Budi	GO	127	913	3	Fall	2009	D	454	M
90774	113	1	Spring	2006	C+	90774	Yohan	HH	24	113	1	Spring	2006	H	653	C
90774	536	1	Fall	2008	A	90774	Yohan	HH	24	536	1	Fall	2008	F	352	H
90774	536	1	Fall	2008	A	90774	Yohan	HH	24	536	2	Spring	2010	D	454	F
90774	934	1	Spring	2010	B	90774	Yohan	HH	24	934	1	Spring	2010	D	499	F
91029	227	1	Fall	2005	A	91029	Ahmad	RQ	128	227	1	Fall	2005	G	239	A
91029	227	1	Fall	2005	A	91029	Ahmad	RQ	128	227	2	Spring	2005	J	334	C
91029	729	1	Spring	2004	B-	91029	Ahmad	RQ	128	729	1	Spring	2004	F	352	A
95165	363	1	Fall	2001	A-	95165	Adri	DF	116	363	1	Fall	2001	A	517	A
96384	381	1	Fall	2006	C+	96384	yuyun	GO	5	381	1	Fall	2006	J	334	O
96384	436	1	Fall	2003	A-	96384	yuyun	GO	5	436	1	Fall	2003	G	239	I
96999	160	1	Fall	2006	A-	96999	Kiki	RR	56	160	1	Fall	2006	J	334	J
96999	282	2	Fall	2004	B	96999	Kiki	RR	56	282	1	Fall	2009	D	583	C
96999	282	2	Fall	2004	B	96999	Kiki	RR	56	282	2	Fall	2004	D	454	A
96999	561	1	Spring	2009	A+	96999	Kiki	RR	56	561	1	Spring	2009	D	583	C
96999	561	1	Spring	2009	A+	96999	Kiki	RR	56	561	2	Spring	2005	D	454	L
97383	113	1	Spring	2006	C-	97383	Yohan	EL	58	113	1	Spring	2006	H	653	C
97383	491	3	Spring	2005	A-	97383	Yohan	EL	58	491	1	Fall	2006	H	653	O
97383	491	3	Spring	2005	A-	97383	Yohan	EL	58	491	2	Spring	2004	G	239	P
97383	491	3	Spring	2005	A-	97383	Yohan	EL	58	491	3	Spring	2005	D	499	A
97383	727	1	Fall	2003	B	97383	Yohan	EL	58	727	1	Fall	2003	J	334	D
97383	727	1	Fall	2003	B	97383	Yohan	EL	58	727	2	Fall	2005	J	334	E
97764	113	1	Spring	2006	A+	97764	Adri	ED	48	113	1	Spring	2006	H	653	C
97764	714	1	Spring	2003	C+	97764	Adri	ED	48	714	1	Spring	2003	J	334	G
97764	714	1	Spring	2003	C+	97764	Adri	ED	48	714	2	Spring	2006	D	499	E
97764	938	1	Fall	2001	A+	97764	Adri	ED	48	938	1	Fall	2001	D	499	C
97764	952	2	Fall	2010	A	97764	Adri	ED	48	952	1	Spring	2008	F	352	M
97764	952	2	Fall	2010	A	97764	Adri	ED	48	952	2	Fall	2010	D	454	K
98308	727	1	Fall	2003	A+	98308	Budi	RR	129	727	1	Fall	2003	J	334	D
98308	727	1	Fall	2003	A+	98308	Budi	RR	129	727	2	Fall	2005	J	334	E
99273	171	1	Fall	2009	C+	99273	Yohan	RR	12	171	1	Fall	2009	F	352	H
99273	468	1	Fall	2003	C+	99273	Yohan	RR	12	468	1	Fall	2003	D	583	L

363 rows in set (0.00 sec)

Command Prompt - mysql -u root

Josu	WW	5	Spring	454	D	955	WW
Josu	WW	5	Spring	454	D	955	WW
Adri	GO	1	Fall	499	D	938	DE
Adri	GO	1	Fall	352	F	978	WW
Ande	DF	1	Fall	583	D	834	RR
Ande	DF	1	Fall	499	D	834	RR
Ande	DF	1	Fall	454	D	834	RR
Kiki	RQ	1	Spring	352	F	311	DE
rahmat	HH	2	Spring	352	F	582	GO
rahmat	HH	2	Spring	239	G	582	GO
rahmat	HH	1	Spring	454	D	697	HH
rahmat	HH	1	Spring	454	D	787	DE
rahmat	HH	1	Spring	239	G	787	DE
Johan	WW	3	Spring	352	F	201	DK
Johan	WW	3	Spring	583	D	201	DK
Johan	WW	3	Spring	583	D	201	DK
yuyun	EL	1	Fall	346	J	661	HH
Ande	WW	1	Spring	334	J	578	HH
Ande	WW	1	Spring	454	D	697	HH
Ande	WW	2	Spring	517	A	702	RR
Ande	WW	2	Spring	104	I	702	RR
Ande	WW	2	Spring	454	D	787	DE
Ande	WW	2	Spring	239	G	787	DE
Budi	GO	1	Spring	104	I	116	WW
Budi	GO	2	Fall	454	D	546	ED
Budi	GO	2	Fall	104	I	546	ED
Budi	GO	3	Fall	334	J	913	DK
Budi	GO	3	Fall	346	J	913	DK
Budi	GO	3	Fall	454	D	913	DK
Yohan	HH	1	Spring	653	H	113	WW
Yohan	HH	1	Fall	352	F	536	GO
Yohan	HH	1	Fall	454	D	536	GO
Yohan	HH	1	Spring	499	D	934	ED
Ahmad	RQ	1	Fall	239	G	227	WW
Ahmad	RQ	1	Fall	334	J	227	WW
Ahmad	RQ	1	Spring	352	F	729	DE
Adri	DF	1	Fall	517	A	363	DE
yuyun	GO	1	Fall	334	J	381	RQ
yuyun	GO	1	Fall	239	G	436	ED
Kiki	RR	1	Fall	334	J	160	DE
Kiki	RR	2	Fall	583	D	282	ED
Kiki	RR	2	Fall	454	D	282	ED
Kiki	RR	1	Spring	583	D	561	DF
Kiki	RR	1	Spring	454	D	561	DF
Yohan	EL	1	Spring	653	H	113	WW
Yohan	EL	3	Spring	653	H	491	RQ
Yohan	EL	3	Spring	239	G	491	RQ
Yohan	EL	3	Spring	499	D	491	RQ
Yohan	EL	1	Fall	334	J	727	ED
Yohan	EL	1	Fall	334	J	727	ED
Adri	ED	1	Spring	653	H	113	WW
Adri	ED	1	Spring	334	J	714	WW
Adri	ED	1	Spring	499	D	714	WW
Adri	ED	1	Fall	499	D	938	DE
Adri	ED	2	Fall	352	F	952	DK
Adri	ED	2	Fall	454	D	952	DK
Budi	RR	1	Fall	334	J	727	ED
Budi	RR	1	Fall	334	J	727	ED
Yohan	RR	1	Fall	352	F	171	RQ
Yohan	RR	1	Fall	583	D	468	HH

363 rows in set (0.00 sec)