Essential Skills - RDBMS and SQL

Essential Skills
RDBMS and SQL

Daniël van Eeden
dveeden@snow.nl

October 2011

snow
UNIX SPECIALISTS

What is a Database?

- A structured collection of data

What is a DBMS

- **D**ata**B**ase **M**anagement **S**ystem
- A system which allows you to store data in a generic way.

RDBMS History

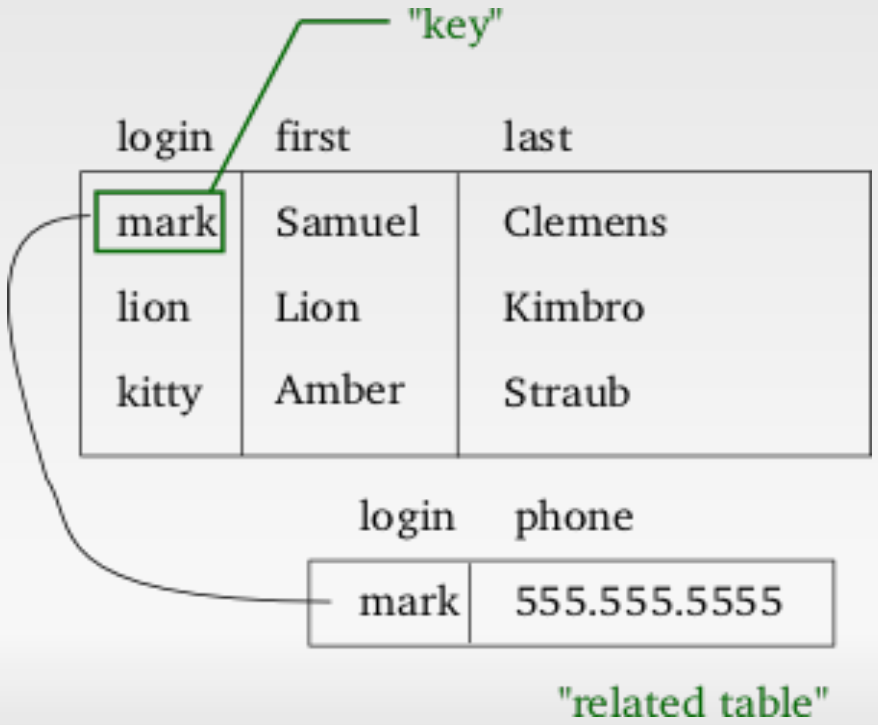
- Relational Database
- Based on a paper by Edgar F. Codd in 1970
- Allowed to take advantage of disk based storage.
- Earlier DBMS implementations forced the programmer to do a full scan over the data, which is okay for tape based storage.

Why use a RDBMS?

- You don't have to write your own datastore
- Allows concurrent access
- Enforce quality of data
  - Constraints
  - Column types
- Makes your data searchable

The relational model

- Store data in tables
- Normalization of data
- Link tables together with keys



ACID

- **A**tomicity
  - Something happens completely or not at all
- **C**onsistency
  - Only consistent data will be written to the database
- **I**solation
  - Transactions are isolated from each other
  - Ensures reliable outcomes
- **D**urability
  - Data is protected against crashes

SQL

- **S**tructured **Q**uery **L**anguage
- **DDL** - Data Definition Language
  - CREATE TABLE, ALTER TABLE
- **DML** - Data Manipulation Language
  - INSERT, UPDATE, DELETE, SELECT
- **DCL** - Data Control Languge
  - GRANT, REVOKE
- **TCL** - Transaction Control Language
  - START TRANSACTION, COMMIT, ROLLBACK

SQL

- Declarative Language
- Many standards
  - SQL-92, SQL:2003, SQL:2006,…
  - ANSI, FIPS, ISO/IEC
- Divided into parts
  - SQL/MED
  - SQL/PSM
  - SQL/XML
  - …

Three-valued Logic

- A column can have three different kind of values:

1. A value: “some string” or 25
2. An empty or zero value: “” or 0
3. No value: NULL

Foreign Keys

- Foreign Keys are relations between tables
- Your database will protect the relationship by enforcing the foreign key. This is called a foreign key constraint.

Data Definition Language

```
test=# CREATE TABLE "user" (uid SERIAL PRIMARY KEY, "name" VARCHAR(255));

NOTICE:  CREATE TABLE will create implicit sequence "user_uid_seq" for serial column
"user.uid"

NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "user_pkey" for table "user"

CREATE TABLE

test=# CREATE TABLE "group" (gid SERIAL PRIMARY KEY, "name" VARCHAR(255));

NOTICE:  CREATE TABLE will create implicit sequence "group_gid_seq" for serial column
"group.gid"

NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "group_pkey" for table
"group"

CREATE TABLE

test=# CREATE TABLE "groupmembers" (mid SERIAL PRIMARY KEY, uid INT REFERENCES "user"
(uid), gid INT REFERENCES "group" (gid));

NOTICE:  CREATE TABLE will create implicit sequence "groupmembers_mid_seq" for serial
column "groupmembers.mid"

NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "groupmembers_pkey" for
table "groupmembers"

CREATE TABLE
```

Data Definition Language

- CREATE
- ALTER
- DROP
- Data Types
  - INT
  - CHAR, VARCHAR
  - BLOB
- Primary Keys, Foreign Keys

Data Manipulation Language

```
test=# INSERT INTO "user" (name) VALUES ('Steve');
INSERT 0 1
test=# INSERT INTO "group" (name) VALUES ('Steve''s Group');
INSERT 0 1
test=# INSERT INTO "groupmembers" (uid,gid) VALUES (1,1);
INSERT 0 1
test=# SELECT * FROM "user";
 uid | name
-----+-------
   1 | Steve
(1 row)
```

Data Manipulation Language

```
test=# UPDATE "user" SET name='Steven' WHERE uid=1;

UPDATE 1

test=# UPDATE "group" SET name='Steven''s Group' WHERE gid=1;

UPDATE 1

test=# DELETE FROM "user" WHERE uid=1;

ERROR:  update or delete on table "user" violates foreign key
constraint "groupmembers_uid_fkey" on table "groupmembers"

DETAIL:  Key (uid)=(1) is still referenced from table
"groupmembers".

test=# DELETE FROM "groupmembers" WHERE uid=1;

DELETE 1

test=# DELETE FROM "user" WHERE uid=1;

DELETE 1
```

Data Manipulation Language

- SELECT, INSERT, UPDATE, DELETE
- WHERE
  - …WHERE fname='Richard' AND lname LIKE 's%'
- Functions
  - SELECT upper(fname) FROM …
  - …WHERE upper(fname) = 'RICHARD'

Data Manipulation Language

- GROUP BY
  - GROUP BY lname,fname
  - SUM(), COUNT(), MAX(), MIN()
- ORDER BY
- LIMIT 100

Joins

```
test=# SELECT u.name Username FROM "user" u LEFT JOIN groupmembers gm ON gm.uid=u.uid
LEFT JOIN "group" g ON g.gid=gm.gid WHERE g.name='Database Administrators';

 username

----------

 George

 Simon

 Patrick

(3 rows)

test=# SELECT g.name,COUNT(*) FROM "group" g LEFT JOIN groupmembers gm ON g.gid=gm.gid
GROUP BY g.name;

          name            | count

--------------------------+-------

 Steven's Group           |     1

 Database Administrators  |     3

 Developers               |     1

 System Administrators    |     1

(4 rows)
```

Transactions

```
test=# START TRANSACTION;
START TRANSACTION
test=# SELECT * FROM groupmembers WHERE uid=4;
 mid | uid | gid
-----+-----+-----
   3 |   4 |   4
   4 |   4 |   5
(2 rows)

test=# DELETE FROM groupmembers WHERE uid=4;
DELETE 2
```

Transactions

```
test=# SELECT * FROM groupmembers WHERE uid=4;
 mid | uid | gid
-----+-----+-----
(0 rows)

test=# ROLLBACK;
ROLLBACK
test=# SELECT * FROM groupmembers WHERE uid=4;
 mid | uid | gid
-----+-----+-----
   3 |   4 |   4
   4 |   4 |   5
(2 rows)
```

Transactions

- START TRANSACTION
- COMMIT
- ROLLBACK
- Transaction Isolation
  - SERIALIZABLE
  - REPEATABLE READ
  - READ COMMITTED
  - READ UNCOMMITTED

MVCC and Transaction Logs

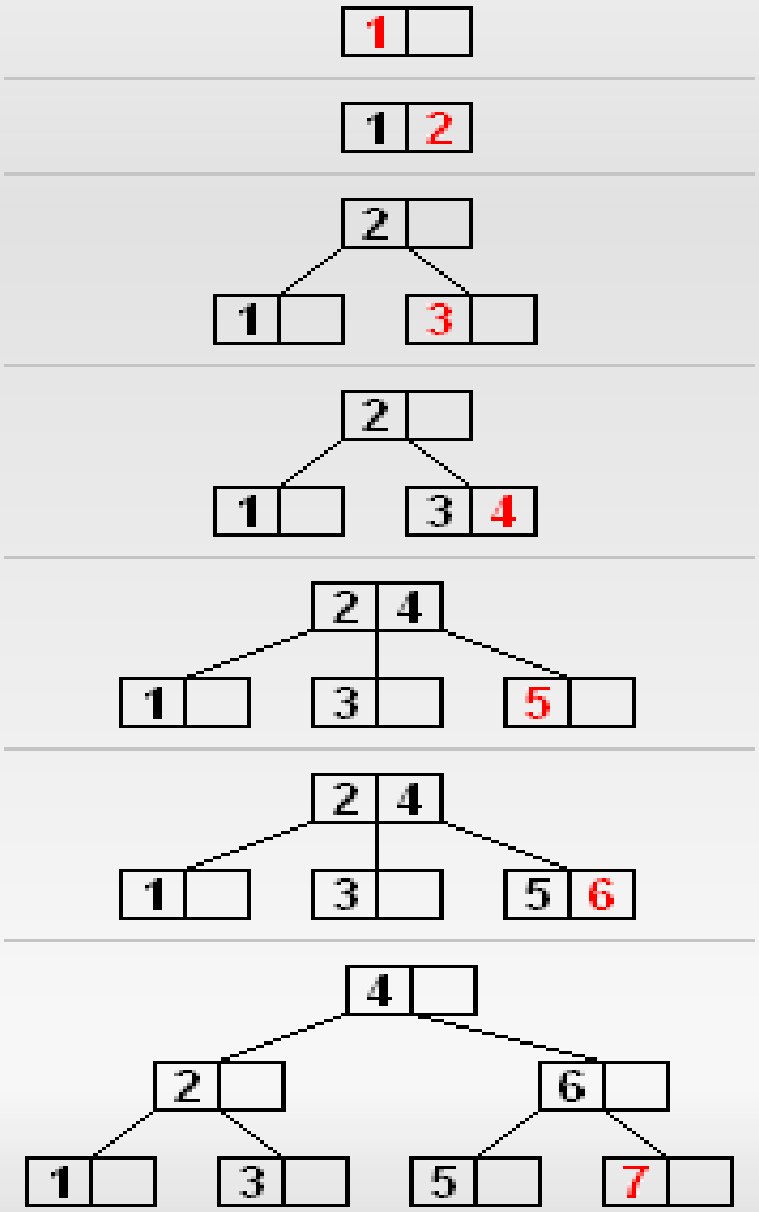
- **M**ulti **V**ersion **C**oncurrency **C**ontrol
- Changes are written to the transaction logs
- This is used for
  - Crash recovery (Similar to Logging filesystems)
  - Online backup
  - MVCC

Indices and the QEP

- An index allows the database to find the needed rows faster.
- It's easy to find someones number in a phonebook.
- It's quite hard to find the name which belongs to a number.
- You can create an index to speedup lookups by number:
  - Number: 575-4444, Page 78

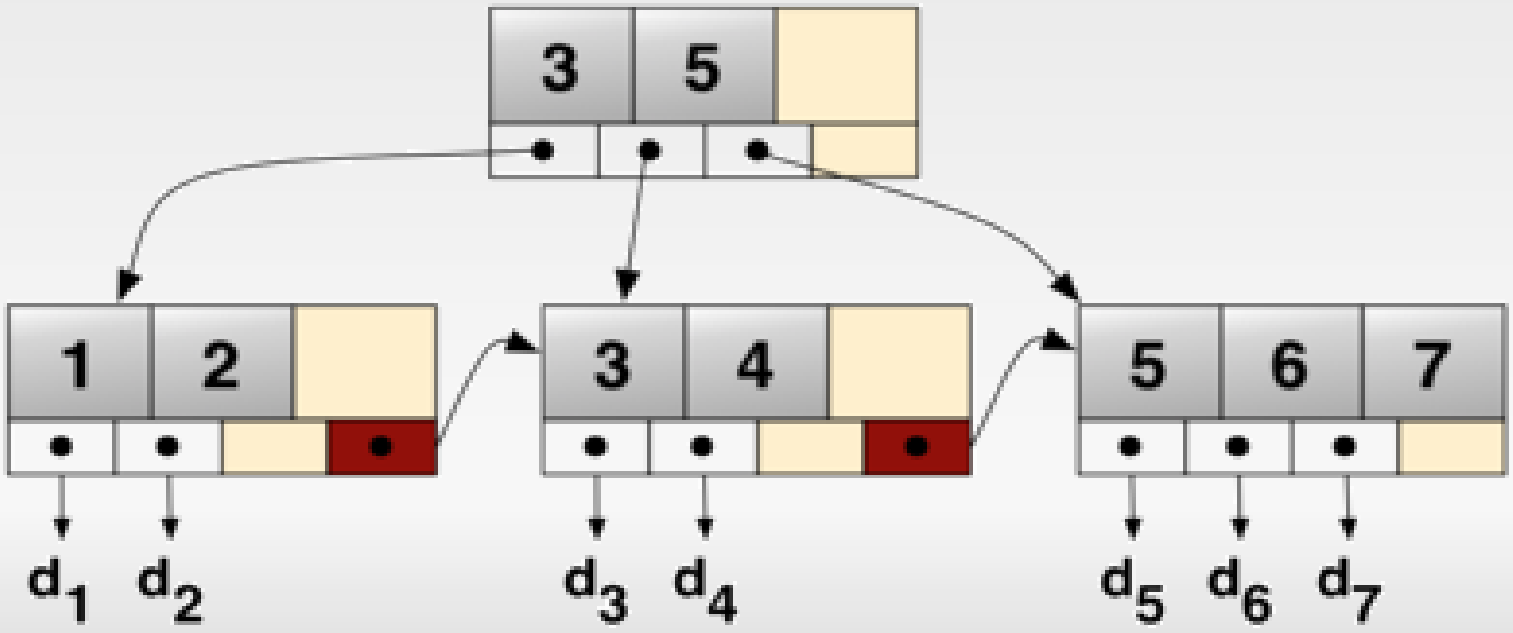
B-Tree

- Used to store tables



B+Tree

- Used to store indices



Cache-Oblivious B-Trees

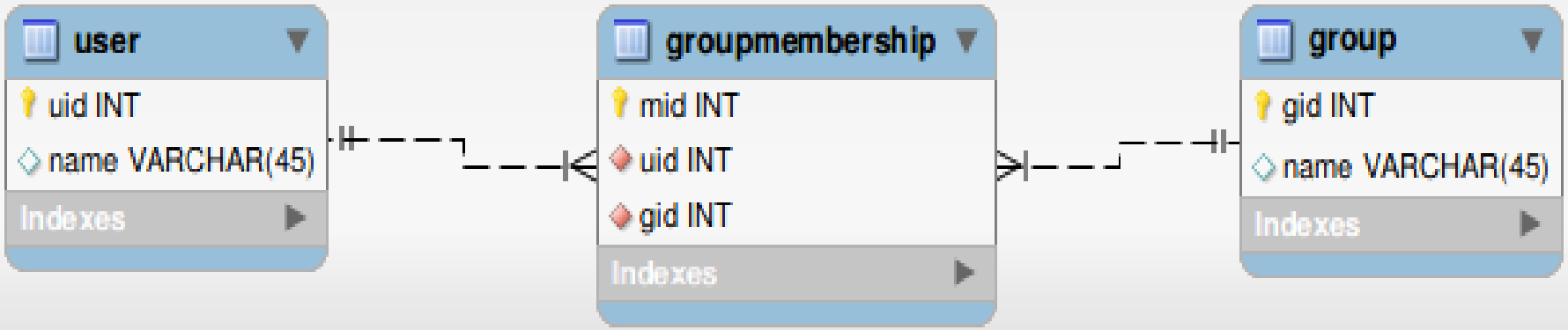
- Implemented as Fractal Tree™ by the TokuDB storage engine for MySQL
- Caches data in non-leaf trees
- Avoids fragmentation
- Allows for high insert speeds

Stored procedures

- Allows you to store business logic in the database
- Allows you to add functions to the database
- Examples:
- `CALL addgroup('System Administators');`
- `SELECT up1(first_name) FROM employees;`

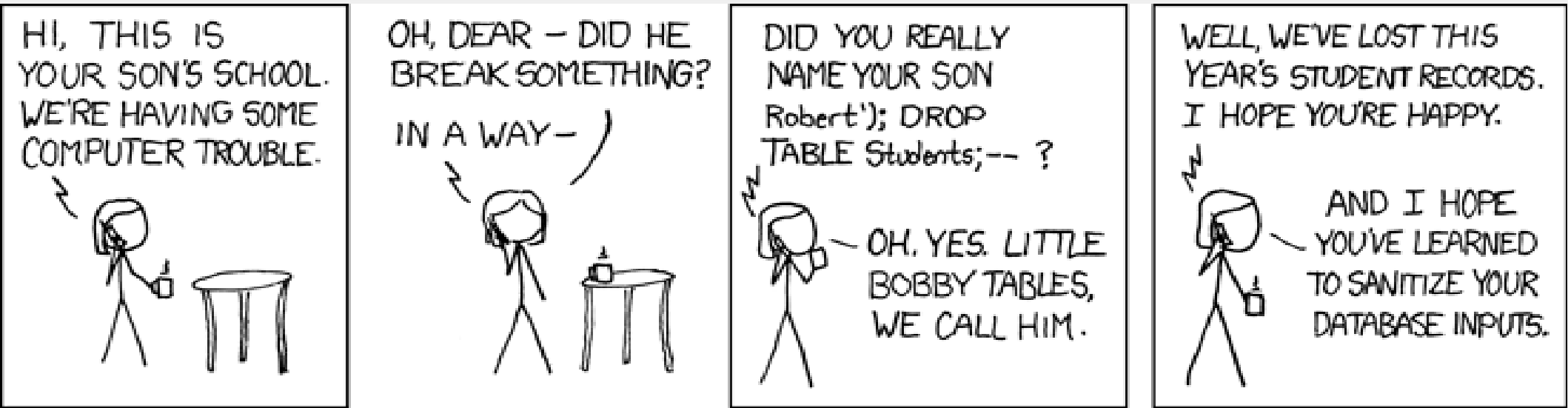
Database design

- ERD - Entity-Relationship Diagram
- Crow's Foot Notation



Using SQL in applications

- Security (SQL Injections)
- ORM - Object Relational Mapping



SQL Injection

- `$sql= "SELECT article_body FROM articles WHERE pageid=$pageid"`
- index.php?pageid=5
- `SELECT article_body FROM articles WHERE pageid=5`
- index.php?pageid=5 OR 1=1; --
- `SELECT article_body FROM articles WHERE pageid=5' OR 1=1; --`

OLTP, OLAP

- Online Analytical Processing
- Online Transaction Processing

Opensource RDBMS

- PostgreSQL
- MySQL
- SQLite

SQLite

- It's in every Firefox and Chrome installation
- It's in every MacOS X installation
- Bundled with PHP and Python
- Solaris 10 needs SQLite to boot
- Every iPhone, Android device and most Symbian phones.
- It's used in Skype and McAfee

SQLite

- serverless, zero-configuration, transactional

MySQL

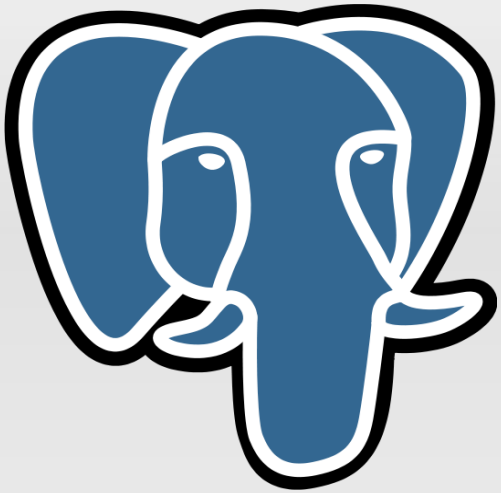
- The M in LAMP
- Multiple storage engines
  - MyISAM, InnoDB, NDB Cluster

MySQL - Alexa Top 20

1. Google
2. Facebook
3. YouTube
4. Yahoo!
5. Baidu.com
6. Wikipedia
7. Blogger.com
8. Windows Live
9. Twitter
10. QQ.COM
11. LinkedIn
12. MSN
13. Yahoo! Japan
14. Amazon.com
15. Google India
16. 淘宝网 (Taobao.com)
17. 新浪新闻中心 (Sina.com.cn)
18. WordPress.com
19. Google.de
20. Google.com.hk

PostgreSQL

- Many features
- Standards compliant



Closed source RDBMS

- Oracle RDBMS
- Sybase ASE
- Microsoft SQL Server
- IBM DB2

NoSQL

Stores data in other formats that tables/rows

- Object stores
- Document stores
- key-value stores

Access data using other protocols/languages than SQL

- Simple text based and/or binary protocols

NoSQL

- Automatic sharding
- Automatic replication
- Eventual consistency

NoSQL Implementations

NoSQL implementations:

- MongoDB
- CouchDB
- Redis
- Cassandra

Hybrid solutions

Allows NoSQL protocol for performance and uses SQL for ad-hoc queries.

- MySQL with HandlerSocket
- MySQL with Memcached
- MySQL with NDB Cluster

Pitfalls

- CAP Theorem
- Scalability
- Locking

Questions?

dveeden@snow.nl

What does a DBA do?

- Check backups and restores
- Create users
- Grant permissions
- Setup Monitoring
- Setup Replication
- Tune the database
- Tune queries
- Add indices
- Rewrite Queries
- Test upgrades
- Benchmark
- Create functions and procedures

Bonus Items

- Views
- Triggers
- Explain
- Subqueries
- Surrogate Keys
- Trees, Multivalue attributes
- LIKE '%
- InnoDB
- BB RAID