MariaDB / MySQL for Web Developers

Web Developer Congress 2020, remote

Oli Sennhauser

Senior MariaDB & MySQL Consultant at FromDual GmbH

https://www.fromdual.com/presentations

About FromDual GmbH

www.fromdual.com

Support

Consulting

remote-DBA

Training

Contents

MariaDB / MySQL for Web Developers

What are databases for?

- Primarily: Relational DBMS (RDBMS)
- Storing Business Information:
  - CRM, ERP, Accounting, Shop, Booking, etc.
- What are they NOT for (non optimal)?
  - Logs → Files, Logstash
  - Images, PDFs, huge texts → Filer, Solr
  - Trash → Waste bin

Different types of databases

- Flat files, CSV, ISAM
- Hierarchical database
- Relational databases (RDBMS)
- Network databases
- Object Oriented databases (OODBMS)
  - Object Relational DBMS (ORDBMS)
- Graph databases
- Column Stores (MariaDB CS)
- "Document" Stores (JSON, MongoDB)
- Wide Column Stores (Cassandra, HBase)

Common Relational DBMS

- MariaDB
  - more in the Web-Client-Server field (LAMP)
- MySQL
  - more in the Web-Client-Server field (LAMP)
- PostgreSQL
  - more in the fat-Client-Server Business Software field
- SQLite
  - Not a real "Client-Server-DBMS" → Library
  - Embedded DBMS (Industry, Firefox, etc.)

Connection to the DBMS

- GUI (MySQL Workbench, HeidiSQL)
- CLI (`mariadb`, `mysql`)
- Watching TV vs. reading
- Connect to the database:
  - User + Password + Host (DNS or IP) + Port
  - User must be created before (`examples.sql`)

```
shell> mysql --user=app --password ↵
       --host=192.168.1.42 --port=3306
Enter password:
```

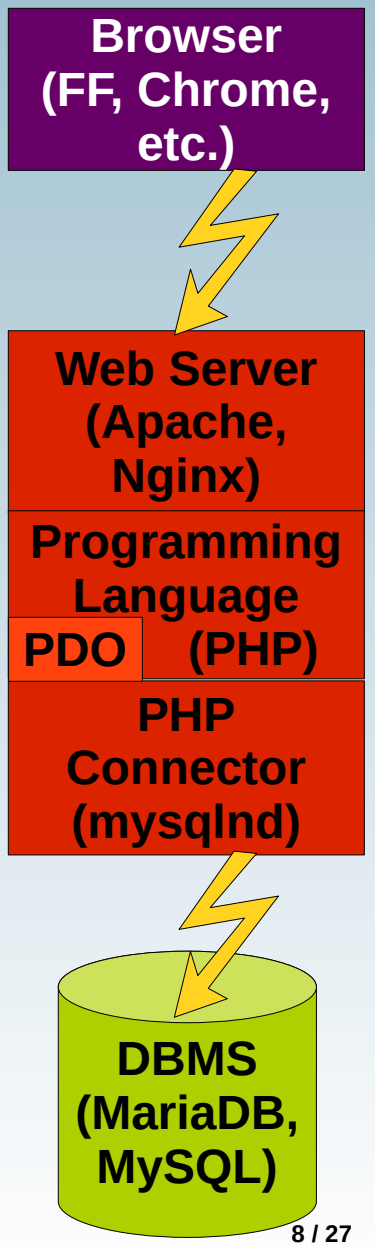
PHP Connector (mysqlnd)

- MySQL Native Driver

```
shell> apt-get install php-mysqlnd

shell> yum install php-mysqlnd
```

  - PDO will be installed as well

- PHP Data Objects (PDO)
  - Data-access abstraction layer
  - For different RDBMS



Connection via PHP

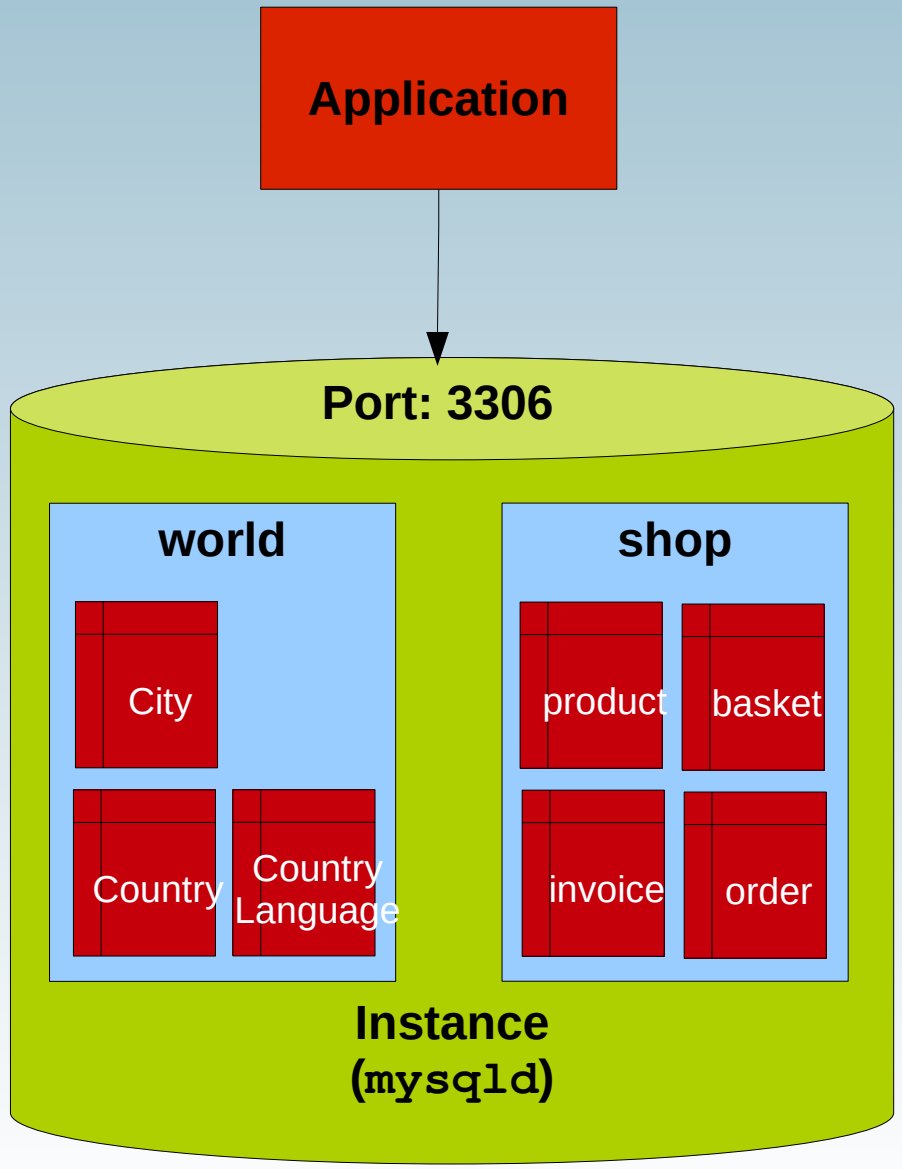

```
$lHost     = '127.0.0.1';
$lPort     = 3306;
$lUser     = 'app';
$lPassword = 'Secret!123';
$lSchema   = 'shop';

$mysqli = new mysqli($lHost, $lUser, $lPassword, $lSchema, $lPort);
// returns DB connection object or false

if ( $mysqli->connect_errno != 0 ) {
  error_log('Error number : ' . $mysqli->connect_errno);
  error_log('Error message: ' . $mysqli->connect_error);
  // Possibly exit here?
  exit(42);
}

// ...

$mysqli->close();
exit(0);
```

Rough structure of a RDBMS

- **MariaDB / MySQL:**

- `SQL> SHOW SCHEMAS;`
- `SQL> use <schema>;`

- `SQL> SHOW TABLES;`
- `SQL> DESC <table>;`



Basic database queries

- SQL - Structured Query Language
- Table similar to a spread sheet (rows and columns)

- Query data:

```
SQL> SELECT * FROM <table>;
SQL> SELECT col1, col3, col4 FROM <table>;
SQL> SELECT * FROM <table>
     WHERE col2 = <value>;
SQL> ... ORDER BY col1 ASC, col2, DESC;
SQL> ... GROUP BY col1, col2;
```

Basic SELECT Query


```
SQL> SELECT * FROM product;
+----+-------+------------+-------+-------+
| id | name  | category   | price | stock |
+----+-------+------------+-------+-------+
|  1 | Apple | fruits     |  3.00 |     5 |
|  2 | Salad | vegetables |  1.00 |     2 |
|  3 | Melon | fruits     |  4.00 |     2 |
|  4 | Onion | vegetables |  0.75 |    10 |
|  5 | Pear  | fruits     |  3.25 |     6 |
|  6 | Bread | pastries   |  1.25 |     4 |
+----+-------+------------+-------+-------+
```

- `SQL> SELECT name, price, stock FROM product;`

	A	B	C	D	E
1	id	name	category	price	stock
2	1	Apple	fruits	3.00	5
3	2	Salad	vegetables	1.00	2
4	3	Melon	fruits	4.00	2
5	4	Onion	vegetables	0.75	10
6	5	Pear	fruits	3.25	6
7	6	Bread	pastries	1.25	4

Filter with WHERE

- `SQL> SELECT * FROM product`
  `WHERE category = 'fruits';`

	A	B	C	D	E
1	id	name	category	price	stock
2	1	Apple	fruits	3.00	5
3	2	Salad	vegetables	1.00	2
4	3	Melon	fruits	4.00	2
5	4	Onion	vegetables	0.75	10
6	5	Pear	fruits	3.25	6
7	6	Bread	pastries	1.25	4

```
SQL> SELECT name, price, stock
       FROM product
      WHERE category = 'fruits';
+-------+-------+-------+
| name  | price | stock |
+-------+-------+-------+
| Apple |  3.00 |     5 |
| Melon |  4.00 |     2 |
| Pear  |  3.25 |     6 |
+-------+-------+-------+
```

Sorting and Aggregating

- `SQL> ... ORDER BY stock DESC, price ASC;`
- `SQL> ... GROUP BY col1, col2;`

```
+-------+-------+-------+
| name  | price | stock |
+-------+-------+-------+
| Onion |  0.75 |    10 |
| Pear  |  3.25 |     6 |
| Apple |  3.00 |     5 |
| Bread |  1.25 |     4 |
| Salad |  1.00 |     2 |
| Melon |  4.00 |     2 |
+-------+-------+-------+
```

```
SQL> SELECT category
          , SUM(stock) AS count
       FROM product
      GROUP BY category;
+------------+-------+
| category   | count |
+------------+-------+
| fruits     |    13 |
| pastries   |     4 |
| vegetables |    12 |
+------------+-------+
```

Query data in PHP


```
$sql = 'SELECT category, SUM(stock) AS count FROM product GROUP BY category';

if ( $result = $mysqli->query($sql) ) {

  printf("Result set has %d rows.\n", $result->num_rows);

  while ( $row = $result->fetch_assoc() ) {
    printf("%s: %d\n", $row['category'], $row['count']);
  }
  $result->close();
}
// $result === false
else {
  error_log(sprintf("Error number : %d\n", $mysqli->errno));
  error_log(sprintf("Error message: %s\n", $mysqli->error));
}
```

Changing Data (DML)

- DML – Data Manipulation Language
- **INSERT, UPDATE, DELETE**

```
SQL> INSERT INTO product (name, category, price, stock)
     VALUES ('Salmon', 'seafood', 20.00, 1);
```

```
SQL> UPDATE product
        SET stock = stock - 1
     -- This is Salad!
      WHERE id = 2;
```

	A	B	C	D	E
1	id	name	category	price	stock
2	1	Apple	fruits	3.00	5
3	2	Salad	vegetables	1.00	2
4	3	Melon	fruits	4.00	2
5	4	Onion	vegetables	0.75	10
6	5	Pear	fruits	3.25	6
7	6	Bread	pastries	1.25	4

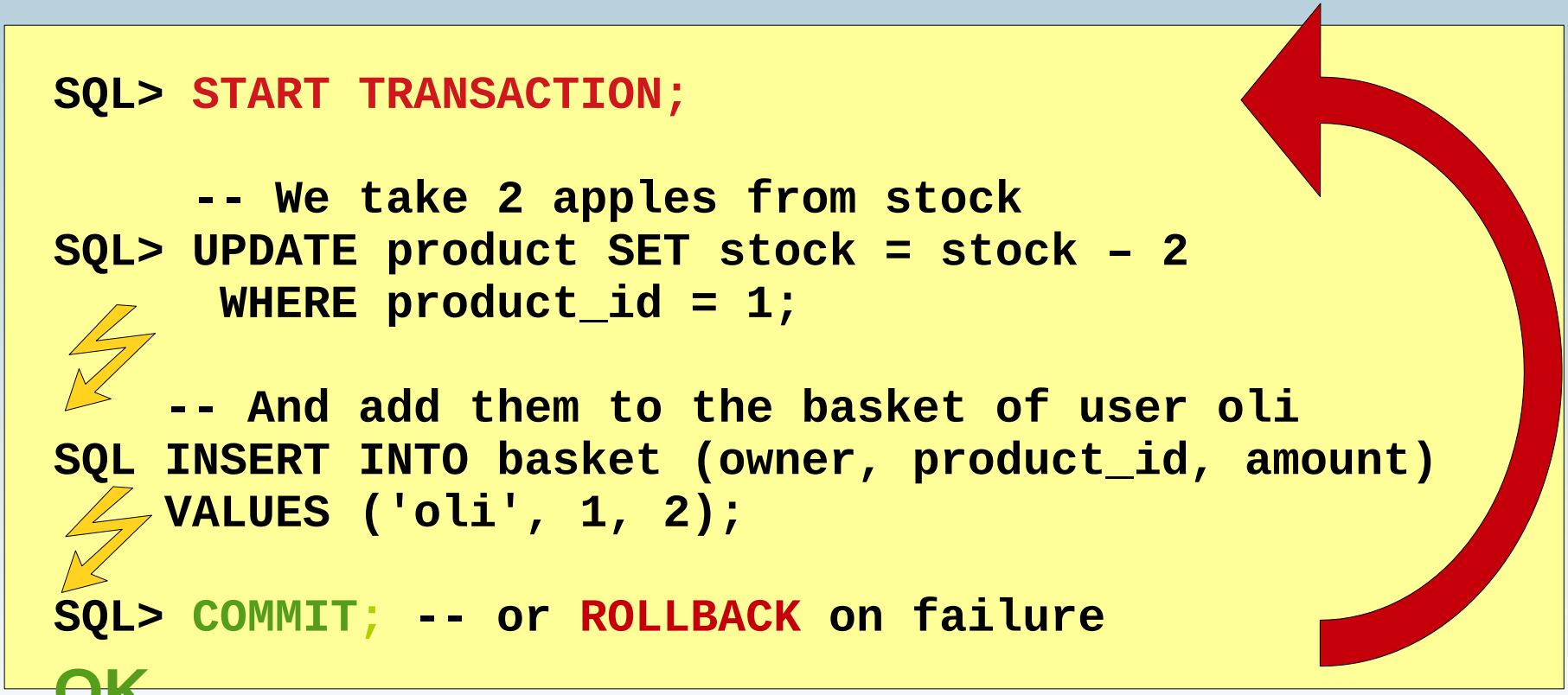
```
SQL> DELETE FROM product
      WHERE category = 'pastries';
```

Changing data in PHP


```
$dml = 'UPDATE product SET stock = stock - 1 WHERE id = 2';

if ( $result = $mysqli->query($dml) ) {

    printf("DML statement altered %d rows.\n",  $mysqli->affected_rows);
}
// $result === false
else {
  error_log(sprintf("Error number : %d\n", $mysqli->errno));
  error_log(sprintf("Error message: %s\n", $mysqli->error));
}
```

Transactions (trx)

- **Adding several DML statement into ONE business operation (= Transaction)**



Transactions in PHP


```
try {

  $mysqli->begin_transaction();
  ...
  $mysqli->query($dml1);
  $mysqli->query($dml2);
  ...
  $mysqli->commit();
}
catch ( mysqli_sql_exception $exception ) {
  $mysqli->rollback();
}
```

PHP - Errors

- On Connect

```
$mysqli = new mysqli($lHost, $lUser, $lPassword, $lSchema, $lPort);
// returns DB connection object or false

if ( $mysqli->connect_errno != 0 ) {
  error_log('Error number : ' . $mysqli->connect_errno);
  error_log('Error message: ' . $mysqli->connect_error);
  // Possibly exit here?
  exit(42);
}
```

- On Query

```
if ( ($result = $mysqli->query($sql)) === false ) {
  error_log(sprintf("Error number : %d\n", $mysqli->errno));
  error_log(sprintf("Error message: %s\n", $mysqli->error));
  exit(42);
}
```

DB side Debugging

- MariaDB / MySQL Error Log
  - NO application errors but database errors!!!
  - my.cnf:
    ```
    log_error = /var/log/mysql/error.log
    ```
- MariaDB: SQL Error Log Plugin
  - → Application Errors
- MariaDB / MySQL General Log
  - my.cnf:
    ```
    general_log_file = /var/log/mysql/general.log
    ```
  - `SQL> SET GLOBAL general_log = {ON | OFF}`

PHP logging

- **php.ini:**
  - **Requires Apache reload!**

```
php> ini_get('error_log');
php> ini_set('error_log', ...)
```

Directives	Default	Prod
display_errors	1	0
error_log	0	<path>
error_reporting	null	E_ALL
log_errors	0	1

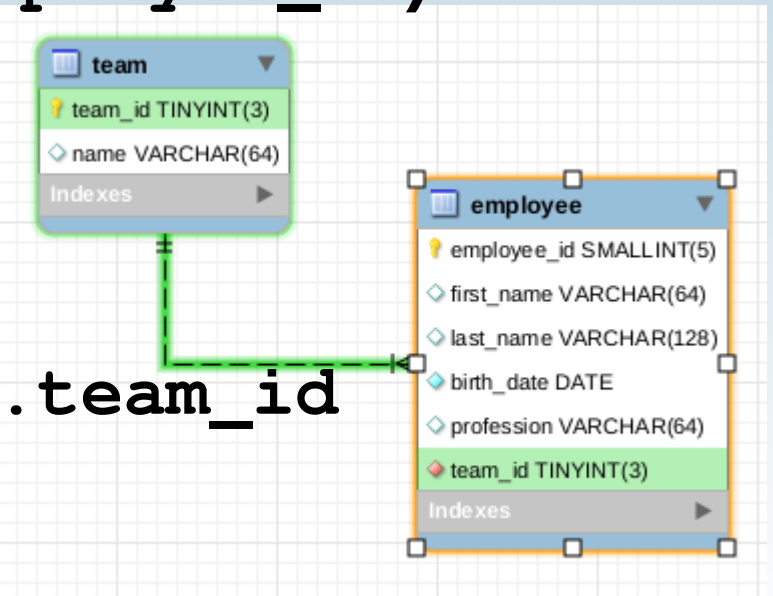
- **Default error log:**
  - `/var/log/apache2/error.log`

```
php> error_log('Your error message here.');

php> syslog(LOG_WARNING, 'Your syslog message here.');
```

Literature: https://www.loggly.com/ultimate-guide/php-logging-basics/

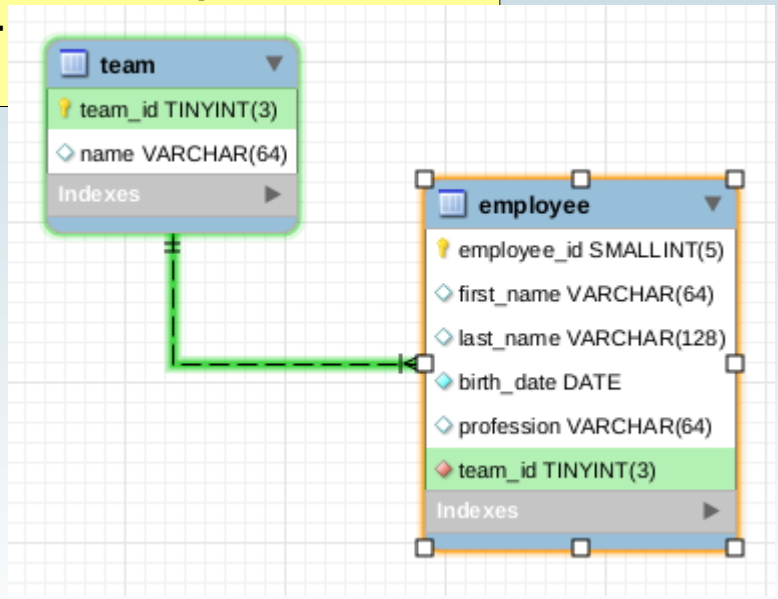
Joining Tables

- Tables have (parent-child) relations:
  - 1 Team (parent) has many (n) Employees (children)
  - 1 employee belongs to 1 team
- Every row has a unique row identifier
  - → Primary Key (`team_id, employee_id`)
- In a relation we have to indicate which child belongs to which parent
  - `employee.team_id → team.team_id`



SQL JOIN Query


```
SELECT team.*, employee.first_name, employee.last_name
  FROM team
  JOIN employee ON employee.team_id = team.team_id
 WHERE employee.first_name = 'oli';

+---------+------------+------------+------------+
| team_id | name       | first_name | last_name  |
+---------+------------+------------+------------+
|       1 | Consulting | Oli        | Sennhauser |
|       5 | Sales      | Oli        | Meier      |
+---------+------------+------------+------------+
```



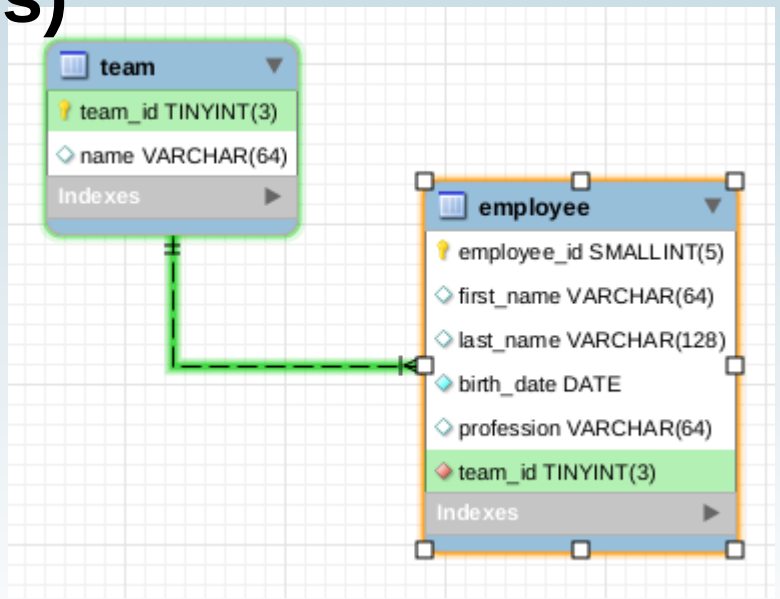
Indexing

- What is an index for?

- Where to put an index?
  - Unique row identifier (Primary Key: `team_id, employee_id`)
  - Columns you `JOIN` on (Foreign Key: `employee.team_id`)
  - Columns you filter on (`WHERE last_name = 'Meier'`)
  - Indices on several columns are possible:
    - `ALTER TABLE employee ADD INDEX (last_name, first_name);`

Create your own Schema

- Entity Relationship (ER) diagram
- Entity → Thing (`team`, `employee`)
- Relationship → Relations between things
- Nomalization → fragement "world" to model into entities (no more redundancies)
  - Each column of table must be atomic (name → first + last)
  - Synthetic Primary Keys
  - Do not mix 2 different things in one table
  - Avoid redundancies



Q & A

Questions ?

Discussion?

We have time for some face-to-face talks...

- **FromDual provides neutral and independent:**
  - **Consulting**
  - **remote-DBA**
  - **Support for MariaDB, MySQL and Galera Cluster**
  - **Training**