1. Deploy test environment with Docker

1.1 Start SQL with Docker

docker run --name=mysql1 -d mysql/mysql-server:5.6

docker logs mysql1 2>&1 | grep -i GENERATED

docker exec -it mysql1 mysql -uroot -p

1.2 Initial settings

mysql>use mysql

mysql>set password for root@localhost=password('passw0rd')

mysql>SELECT user,authentication\_string,plugin,host FROM mysql.user;

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

| user | authentication\_string | plugin | host |

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

| root | | mysql\_native\_password | localhost |

| healthchecker | | mysql\_native\_password | localhost |

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

1.3 Import test tables

docker exec -it mysql1 bash

mysql>CREATE DATABASE learning;

Query OK, 1 row affected (0.02 sec)

mysql>use learning;

Database changed

Download scripts from http://www.forta.com/books/0672327120.

mysql> source /script/create.sql

mysql> source /script/insert.sql

1.4 Check databases and tables

mysql> show databases;

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

| Database |

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

| information\_schema |

| learning |

| mysql |

| performance\_schema |

| test |

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

5 rows in set (0.00 sec)

mysql> show tables;

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

| Tables\_in\_learning |

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

| customers |

| orderitems |

| orders |

| productnotes |

| products |

| vendors |

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

6 rows in set (0.00 sec)

mysql> show columns from customers;

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

| Field | Type | Null | Key | Default | Extra |

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

| cust\_id | int(11) | NO | PRI | NULL | auto\_increment |

| cust\_name | char(50) | NO | | NULL | |

| cust\_address | char(50) | YES | | NULL | |

| cust\_city | char(50) | YES | | NULL | |

| cust\_state | char(5) | YES | | NULL | |

| cust\_zip | char(10) | YES | | NULL | |

| cust\_country | char(50) | YES | | NULL | |

| cust\_contact | char(50) | YES | | NULL | |

| cust\_email | char(255) | YES | | NULL | |

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

9 rows in set (0.02 sec)

mysql> show create table customers;

2. SELECT

2.1 Distinct

mysql> SELECT **DISTINCT** vend\_id, prod\_price

-> FROM products;

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

| vend\_id | prod\_price |

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

| 1001 | 5.99 |

| 1001 | 9.99 |

| 1001 | 14.99 |

| 1003 | 13.00 |

| 1003 | 10.00 |

| 1003 | 2.50 |

| 1002 | 3.42 |

| 1005 | 35.00 |

| 1005 | 55.00 |

| 1002 | 8.99 |

| 1003 | 50.00 |

| 1003 | 4.49 |

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

12 rows in set (0.00 sec)

2.2 Order By and Limit

mysql> SELECT prod\_id, prod\_price, prod\_name

-> FROM products

-> **ORDER BY** prod\_price **DESC**, prod\_name;

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

| prod\_id | prod\_price | prod\_name |

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

| JP2000 | 55.00 | JetPack 2000 |

| SAFE | 50.00 | Safe |

| JP1000 | 35.00 | JetPack 1000 |

| ANV03 | 14.99 | 2 ton anvil |

| DTNTR | 13.00 | Detonator |

| FB | 10.00 | Bird seed |

| TNT2 | 10.00 | TNT (5 sticks) |

| ANV02 | 9.99 | 1 ton anvil |

| OL1 | 8.99 | Oil can |

| ANV01 | 5.99 | .5 ton anvil |

| SLING | 4.49 | Sling |

| FU1 | 3.42 | Fuses |

| FC | 2.50 | Carrots |

| TNT1 | 2.50 | TNT (1 stick) |

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

14 rows in set (0.01 sec)

mysql> SELECT prod\_price

-> FROM products

-> **ORDER BY** prod\_price DESC

-> **LIMIT** 1;

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

| prod\_price |

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

| 55.00 |

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

1 row in set (0.00 sec)

2.3 Between And

mysql> SELECT prod\_name, prod\_price

-> FROM products

-> WHERE prod\_price **BETWEEN** 5 **AND** 10;

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

| prod\_name | prod\_price |

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

| .5 ton anvil | 5.99 |

| 1 ton anvil | 9.99 |

| Bird seed | 10.00 |

| Oil can | 8.99 |

| TNT (5 sticks) | 10.00 |

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

5 rows in set (0.02 sec)

2.4 Is Null

mysql> SELECT cust\_id

-> FROM customers

-> WHERE cust\_email **IS NULL**;

+---------+

| cust\_id |

+---------+

| 10002 |

| 10005 |

+---------+

2 rows in set (0.01 sec)

2.5 In and Not In

mysql> SELECT prod\_name, prod\_price

-> FROM products

-> WHERE vend\_id **NOT IN** (1002,1003)

-> ORDER BY prod\_name;

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

| prod\_name | prod\_price |

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

| .5 ton anvil | 5.99 |

| 1 ton anvil | 9.99 |

| 2 ton anvil | 14.99 |

| JetPack 1000 | 35.00 |

| JetPack 2000 | 55.00 |

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

5 rows in set (0.09 sec)

2.6 Like

mysql> SELECT prod\_id, prod\_name

-> FROM products

-> WHERE prod\_name **LIKE** 'jet**%**';

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

| prod\_id | prod\_name |

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

| JP1000 | JetPack 1000 |

| JP2000 | JetPack 2000 |

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

2 rows in set (0.01 sec)

mysql> SELECT prod\_id, prod\_name

-> FROM products

-> WHERE prod\_name **LIKE** '**\_** ton anvil';

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

| prod\_id | prod\_name |

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

| ANV02 | 1 ton anvil |

| ANV03 | 2 ton anvil |

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

2 rows in set (0.00 sec)

2.7 Regexp

mysql> SELECT prod\_name

-> FROM products

-> WHERE prod\_name **REGEXP** '[[:digit:]]{4}';

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

| prod\_name |

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

| JetPack 1000 |

| JetPack 2000 |

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

2 rows in set (0.00 sec)

2.8 As and fucntions

mysql> SELECT **Concat(Rtrim(vend\_name), '(', Rtrim(vend\_country), ')') AS vend\_title**

-> FROM vendors

-> ORDER BY vend\_name;

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

| vend\_title |

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

| ACME(USA) |

| Anvils R Us(USA) |

| Furball Inc.(USA) |

| Jet Set(England) |

| Jouets Et Ours(France) |

| LT Supplies(USA) |

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

6 rows in set (0.01 sec)

mysql> SELECT prod\_id, quantity, item\_price, **quantity\*item\_price AS expanded\_price**

-> FROM orderitems

-> WHERE order\_num = 20005;

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

| prod\_id | quantity | item\_price | expanded\_price |

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

| ANV01 | 10 | 5.99 | 59.90 |

| ANV02 | 3 | 9.99 | 29.97 |

| TNT2 | 5 | 10.00 | 50.00 |

| FB | 1 | 10.00 | 10.00 |

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

4 rows in set (0.00 sec)

mysql> SELECT cust\_name, cust\_contact

-> FROM customers

-> WHERE **Soundex**(cust\_contact) = **Soundex**('Y Lie');

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

| cust\_name | cust\_contact |

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

| Coyote Inc. | Y Lee |

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

1 row in set (0.02 sec)

mysql> SELECT cust\_id, order\_num

-> FROM orders

-> WHERE **Date**(order\_date) BETWEEN '2005-09-01' AND '2005-09-30';

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

| cust\_id | order\_num |

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

| 10001 | 20005 |

| 10003 | 20006 |

| 10004 | 20007 |

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

3 rows in set (0.00 sec)

mysql> SELECT cust\_id, order\_num

-> FROM orders

-> WHERE **Year**(order\_date) = 2005 AND **Month**(order\_date) = 9;

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

| cust\_id | order\_num |

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

| 10001 | 20005 |

| 10003 | 20006 |

| 10004 | 20007 |

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

3 rows in set (0.01 sec)

mysql> SELECT **SUM(item\_price\*quantity)** AS total\_price

-> FROM orderitems

-> WHERE order\_num = 20005;

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

| total\_price |

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

| 149.87 |

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

1 row in set (0.04 sec)

2.9 Group By

mysql> SELECT vend\_id, count(\*) AS num\_prods

-> FROM products

-> **GROUP BY** vend\_id **WITH ROLLUP**;

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

| vend\_id | num\_prods |

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

| 1001 | 3 |

| 1002 | 2 |

| 1003 | 7 |

| 1005 | 2 |

| NULL | 14 |

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

5 rows in set (0.01 sec)

2.10 Having

mysql> SELECT vend\_id, COUNT(\*) AS num\_prods

-> FROM products

-> where prod\_price >= 10

-> GROUP BY vend\_id

-> **HAVING** COUNT(\*)>2;

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

| vend\_id | num\_prods |

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

| 1003 | 4 |

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

1 row in set (0.01 sec)

2.11 Join

mysql> SELECT cust\_name, cust\_state, **(SELECT COUNT(\*) FROM orders WHERE orders.cust\_id = customers.cust\_id)**

-> FROM customers

-> ORDER BY cust\_name;

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

| cust\_name | cust\_state | (SELECT COUNT(\*) FROM orders WHERE orders.cust\_id = customers.cust\_id) |

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

| Coyote Inc. | MI | 2 |

| E Fudd | IL | 1 |

| Mouse House | OH | 0 |

| Wascals | IN | 1 |

| Yosemite Place | AZ | 1 |

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

5 rows in set (0.00 sec)

mysql> SELECT p1.prod\_id, p1.prod\_name

-> FROM **products AS p1, products AS p2**

-> WHERE p1.vend\_id = p2.vend\_id AND p2.prod\_id="DTNTR";

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

| prod\_id | prod\_name |

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

| DTNTR | Detonator |

| FB | Bird seed |

| FC | Carrots |

| SAFE | Safe |

| SLING | Sling |

| TNT1 | TNT (1 stick) |

| TNT2 | TNT (5 sticks) |

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

7 rows in set (0.01 sec)

mysql> SELECT c.cust\_id, o.order\_num

-> FROM customers AS c **INNER JOIN** orders AS o

-> ON c.cust\_id = o.cust\_id;

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

| cust\_id | order\_num |

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

| 10001 | 20005 |

| 10001 | 20009 |

| 10003 | 20006 |

| 10004 | 20007 |

| 10005 | 20008 |

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

5 rows in set (0.01 sec)

mysql> SELECT c.cust\_id, o.order\_num

-> FROM customers AS c **LEFT OUTER** JOIN orders AS o

-> ON c.cust\_id = o.cust\_id;

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

| cust\_id | order\_num |

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

| 10001 | 20005 |

| 10001 | 20009 |

| 10002 | NULL |

| 10003 | 20006 |

| 10004 | 20007 |

| 10005 | 20008 |

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

6 rows in set (0.01 sec)

mysql> SELECT c.cust\_id, o.order\_num

-> FROM customers AS c **RIGHT OUTER JOIN** orders AS o

-> ON c.cust\_id = o.cust\_id;

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

| cust\_id | order\_num |

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

| 10001 | 20005 |

| 10001 | 20009 |

| 10003 | 20006 |

| 10004 | 20007 |

| 10005 | 20008 |

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

5 rows in set (0.00 sec)

2.12 Full Text

CREATE TABLE `productnotes` (

`note\_id` int(11) NOT NULL AUTO\_INCREMENT,

`prod\_id` char(10) NOT NULL,

`note\_date` datetime NOT NULL,

`note\_text` text,

PRIMARY KEY (`note\_id`),

**FULLTEXT KEY `note\_text` (`note\_text`)**

) ENGINE=**MyISAM** AUTO\_INCREMENT=115 DEFAULT CHARSET=latin1

mysql> SELECT note\_id, note\_text

-> FROM productnotes

-> WHERE **Match**(note\_text) **Against**('rabbit');

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

| note\_id | note\_text |

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

| 110 | Customer complaint: rabbit has been able to detect trap, food apparently less effective now. |

| 104 | Quantity varies, sold by the sack load.

All guaranteed to be bright and orange, and suitable for use as rabbit bait. |

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

2 rows in set (0.00 sec);

mysql> SELECT note\_id, note\_text

-> FROM productnotes

-> WHERE **Match**(note\_text) **Against**('anvils' **WITH QUERY EXPANSION**);

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

| note\_id | note\_text |

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

| 108 | Multiple customer returns, anvils failing to drop fast enough or falling backwards on purchaser. Recommend that customer considers using heavier anvils. |

| 101 | Customer complaint:

Sticks not individually wrapped, too easy to mistakenly detonate all at once.

Recommend individual wrapping. |

| 113 | Customer complaint:

Not heavy enough to generate flying stars around head of victim. If being purchased for dropping, recommend ANV02 or ANV03 instead. |

| 107 | Please note that no returns will be accepted if safe opened using explosives. |

| 110 | Customer complaint: rabbit has been able to detect trap, food apparently less effective now. |

| 112 | Customer complaint:

Circular hole in safe floor can apparently be easily cut with handsaw. |

| 106 | Matches not included, recommend purchase of matches or detonator (item DTNTR). |

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

7 rows in set (0.01 sec)

mysql> SELECT note\_id, note\_text

-> FROM productnotes

-> WHERE Match(note\_text) Against('rabbit bait' **IN BOOLEAN MODE**);

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

| note\_id | note\_text |

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

| 104 | Quantity varies, sold by the sack load.

All guaranteed to be bright and orange, and suitable for use as rabbit bait. |

| 110 | Customer complaint: rabbit has been able to detect trap, food apparently less effective now. |

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

2 rows in set (0.01 sec)

mysql> SELECT note\_id, note\_text

-> FROM productnotes

-> WHERE Match(note\_text) Against('+rabbit +bait' IN BOOLEAN MODE);

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

| note\_id | note\_text |

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

| 104 | Quantity varies, sold by the sack load.

All guaranteed to be bright and orange, and suitable for use as rabbit bait. |

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

1 row in set (0.00 sec)

mysql> SELECT note\_id, note\_text

-> FROM productnotes

-> WHERE WHERE Match(note\_text) Against('"rabbit bait"' IN BOOLEAN MODE);

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

| note\_id | note\_text |

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

| 104 | Quantity varies, sold by the sack load.

All guaranteed to be bright and orange, and suitable for use as rabbit bait. |

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

1 row in set (0.01 sec)

# 3. INSERT

mysql> INSERT **INTO** customers\_new

-> (cust\_id, cust\_name, cust\_address, cust\_city, cust\_state, cust\_zip, cust\_country, cust\_contact, cust\_email)

-> **VALUES**(10001, 'Coyote Inc.', '200 Maple Lane', 'Detroit', 'MI', '44444', 'USA', 'Y Lee', 'ylee@coyote.com');

Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO customers\_new

-> (cust\_name, cust\_address, cust\_city, cust\_state, cust\_zip, cust\_country, cust\_contact, cust\_email)

-> **SELECT** cust\_name, cust\_address, cust\_city, cust\_state, cust\_zip, cust\_country, cust\_contact, cust\_email

-> FROM customers;

Query OK, 5 rows affected (0.03 sec)

Records: 5 Duplicates: 0 Warnings: 0

# 4. UPDATE

mysql> UPDATE customers

-> **SET** cust\_name= "The Fudds"

-> WHERE cust\_id = '10005';

Query OK, 1 row affected (0.06 sec)

Rows matched: 1 Changed: 1 Warnings: 0

CREATE TABLE orderitems

(

order\_num int NOT NULL ,

order\_item int NOT NULL ,

prod\_id char(10) NOT NULL ,

quantity int NOT NULL ,

item\_price decimal(8,2) NOT NULL ,

PRIMARY KEY (order\_num, order\_item)

) ENGINE=InnoDB;

# 5. FOREIGN KEY

ALTER TABLE orderitems ADD CONSTRAINT fk\_orderitems\_orders FOREIGN KEY (order\_num) REFERENCES orders (order\_num);

# 6. VIEW

mysql> CREATE **VIEW** productcustomers **AS**

-> SELECT cust\_name, cust\_contact, prod\_id

-> FROM customers, orders, orderitems

-> WHERE customers.cust\_id = orders.cust\_id

-> AND orderitems.order\_num = orders.order\_num;

Query OK, 0 rows affecte0pd (0.04 sec)

# 7. PROCEDURE

## 7.1 Simple Procedure

mysql> DELIMITER //

mysql> CREATE **PROCEDURE** productpricing**()**

-> **BEGIN**

-> SELECT Avg(prod\_price) AS priceaverage

-> FROM products;

-> **END** //

Query OK, 0 rows affected (0.18 sec)

mysql> DELIMITER ;

mysql> **CALL** productpricing();

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

| priceaverage |

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

| 16.133571 |

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

1 row in set (0.00 sec)

## 7.2 Procedure with parameters

mysql> CREATE PROCEDURE ordertotal(

-> **IN** onumber INT,

-> **OUT** ototal DECIMAL(8,2)

-> )

-> BEGIN

-> SELECT Sum(item\_price\*quantity)

-> FROM orderitems

-> WHERE order\_num = onumber

-> INTO ototal;

-> END;

mysql> CALL ordertotal(20005, @total);

Query OK, 1 row affected (0.04 sec)

mysql> select @total;

+--------+

| @total |

+--------+

| 149.87 |

+--------+

1 row in set (0.00 sec)

mysql> SHOW CREATE PROCEDURE ordertotal;

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

| Procedure | sql\_mode | Create Procedure | character\_set\_client | collation\_connection | Database Collation |

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

| ordertotal | NO\_ENGINE\_SUBSTITUTION | CREATE DEFINER=`root`@`localhost` PROCEDURE `ordertotal`(

IN onumber INT,

OUT ototal DECIMAL(8,2)

)

BEGIN

SELECT Sum(item\_price\*quantity)

FROM orderitems

WHERE order\_num = onumber

INTO ototal;

END | latin1 | latin1\_swedish\_ci | latin1\_swedish\_ci |

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

1 row in set (0.01 sec)

mysql> DROP PROCEDURE ordertotal;

Query OK, 0 rows affected (0.06 sec)

## 7.3 Declare variables

mysql> CREATE PROCEDURE ordertotal(

-> IN onumber INT,

-> IN taxable BOOLEAN,

-> OUT ototal DECIMAL(8,2)

-> ) COMMENT 'Obtain order total, optionally adding tax'

-> BEGIN

-> -- Declare variable for total

-> DECLARE total DECIMAL(8,2);

-> -- Declare tax percentage

-> DECLARE taxrate INT DEFAULT 6;

-> -- Get the order total

-> SELECT Sum(item\_price\*quantity)

-> FROM orderitems

-> WHERE order\_num = onumber

-> INTO total;

-> -- Is this taxable?

-> IF taxable THEN

-> -- Yes, so add taxrate to the total

-> SELECT total+(total/100\*taxrate) INTO total;

-> END IF;

-> -- And finally, save to our variable

-> SELECT total INTO ototal;

-> END//

Query OK, 0 rows affected (0.00 sec)

mysql> CALL ordertotal(20005,1,@ototal);

Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> select @ototal;

+---------+

| @ototal |

+---------+

| 158.86 |

+---------+

1 row in set (0.00 sec)

8. CURSOR

mysql> DELIMITER //

mysql> CREATE PROCEDURE processorders()

-> BEGIN

-> -- Declare local variables

-> DECLARE done BOOLEAN DEFAULT 0;

-> DECLARE o INT;

-> DECLARE t DECIMAL(8,2);

->

-> -- Declare the cursor

-> DECLARE ordernumbers **cursor**

-> **FOR**

-> SELECT order\_num FROM orders;

->

-> -- Declare continue handler

-> DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done=1;

->

-> -- Create a table to store the result

-> CREATE TABLE IF NOT EXISTS ordertotals

-> (order\_num INT, total DECIMAL(8,2));

->

-> -- Open the cursor

-> **OPEN** ordernumbers;

->

-> -- Loop through all rows;

-> REPEAT

-> -- Get order order number

-> **FETCH** ordernumbers INTO o;

->

-> -- Get the total for this order

-> CALL ordertotal(o,1,t);

->

-> -- Insert order and total into ordertotals

-> INSERT INTO ordertotals(order\_num, total)

-> VALUES(o,t);

->

-> -- End of Loop

-> UNTIL done END REPEAT;

->

-> -- Close the cursor

-> **CLOSE** ordernumbers;

-> END//

Query OK, 0 rows affected (0.02 sec)

mysql> CALL processorders();

OK, 1 row affected (9.21 sec)

mysql>

mysql> SELECT \* FROM ordertotals;

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

| order\_num | total |

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

| 20005 | 158.86 |

| 20009 | 40.78 |

| 20006 | 58.30 |

| 20007 | 1060.00 |

| 20008 | 132.50 |

| 20008 | 132.50 |

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

6 rows in set (0.00 sec)

9. TRIGGER

mysql> CREATE TRIGGER neworder **BEFORE INSERT ON** vendors

-> **FOR EACH ROW** SET NEW.vend\_state = Upper(NEW.vend\_state);

Query OK, 0 rows affected (0.16 sec)

mysql> SELECT \* from ordertotals;

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

| order\_num | total |

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

| 20005 | 158.86 |

| 20009 | 40.78 |

| 20006 | 58.30 |

| 20007 | 1060.00 |

| 20008 | 132.50 |

| 20008 | 132.50 |

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

6 rows in set (0.00 sec)

# 10. TRANSACTION

## 10.1 Rollback

mysql> **START TRANSACTION;**

Query OK, 0 rows affected (0.01 sec)

mysql> DELETE from ordertotals;

Query OK, 6 rows affected (0.02 sec)

mysql> SELECT \* from ordertotals;

Empty set (0.00 sec)

**mysql> ROLLBACK;**

Query OK, 0 rows affected (0.06 sec)

mysql> SELECT \* from ordertotals;

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

| order\_num | total |

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

| 20005 | 158.86 |

| 20009 | 40.78 |

| 20006 | 58.30 |

| 20007 | 1060.00 |

| 20008 | 132.50 |

| 20008 | 132.50 |

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

6 rows in set (0.00 sec)

## 10.2 Commit

**mysql> START TRANSACTION;**

Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM orderitems WHERE order\_num=20008;

Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM orders WHERE order\_num=20008;

Query OK, 1 row affected (0.00 sec)

mysql> **COMMIT**;

Query OK, 0 rows affected (0.05 sec)

11. CHARACTER SET AND COLLATION

mysql> SHOW VARIABLES LIKE 'character%';

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

| Variable\_name | Value |

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

| character\_set\_client | latin1 |

| character\_set\_connection | latin1 |

| character\_set\_database | latin1 |

| character\_set\_filesystem | binary |

| character\_set\_results | latin1 |

| character\_set\_server | latin1 |

| character\_set\_system | utf8 |

| character\_sets\_dir | /usr/share/mysql/charsets/ |

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

8 rows in set (0.06 sec)

mysql> SHOW VARIABLES LIKE 'collation%';

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

| Variable\_name | Value |

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

| collation\_connection | latin1\_swedish\_ci |

| collation\_database | latin1\_swedish\_ci |

| collation\_server | latin1\_swedish\_ci |

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

3 rows in set (0.00 sec)

mysql> CREATE TABLE mytable (

-> column1 INT,

-> column2 VARCHAR(10),

-> column3 VARCHAR(10) **CHARACTER SET** latin1 **COLLATE** latin1\_general\_ci)

-> DEFAULT **CHARACTER SET** hebrew **COLLATE** hebrew\_general\_ci;

Query OK, 0 rows affected (0.39 sec)

12. ACCESS CONTROL

12.1 Create User

mysql> CREATE USER cc IDENTIFIED BY "mypassw0rd";

Query OK, 0 rows affected (0.05 sec)

mysql> SELECT user FROM user;

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

| user |

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

| cc |

| healthchecker |

| root |

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

3 rows in set (0.00 sec)

12.2 Grant and Revoke access

mysql> GRANT SELECT ON learning.\* TO cc;

Query OK, 0 rows affected (0.03 sec)

mysql> SHOW GRANTS FOR cc;

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

| Grants for cc@% |

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

| GRANT USAGE ON \*.\* TO 'cc'@'%' IDENTIFIED BY PASSWORD '\*5F01243B828FE3DA8EC3757F6546080660F2B836' |

| GRANT SELECT ON `learning`.\* TO 'cc'@'%' |

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

2 rows in set (0.01 sec)

mysql> REVOKE SELECT ON learning.\* FROM cc;

Query OK, 0 rows affected (0.01 sec)

12.2 Change Password

mysql> SET PASSWORD FOR cc=Password("newpass");

Query OK, 0 rows affected (0.01 sec)

13. MONITOR

13.1 Process List

mysql> SHOW PROCESSLIST;

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

| Id | User | Host | db | Command | Time | State | Info |

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

| 1459 | root | localhost | learning | Query | 0 | init | SHOW PROCESSLIST |

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

1 row in set (0.01 sec)

13.2 Explain

mysql> EXPLAIN SELECT prod\_name, prod\_price FROM products WHERE vend\_id NOT IN (1002,1003);

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

| id | select\_type | table | type | possible\_keys | key | key\_len | ref | rows | Extra |

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

| 1 | SIMPLE | products | ALL | fk\_products\_vendors | NULL | NULL | NULL | 14 | Using where |

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

1 row in set (0.00 sec)

13.3 Analyze Table Status

mysql> ANALYZE TABLE orders;

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

| Table | Op | Msg\_type | Msg\_text |

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

| learning.orders | analyze | status | OK |

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

1 row in set (0.13 sec)

mysql> CHECK TABLE orders, orderitems;

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

| Table | Op | Msg\_type | Msg\_text |

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

| learning.orders | check | status | OK |

| learning.orderitems | check | status | OK |

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

2 rows in set (0.01 sec)

-END-