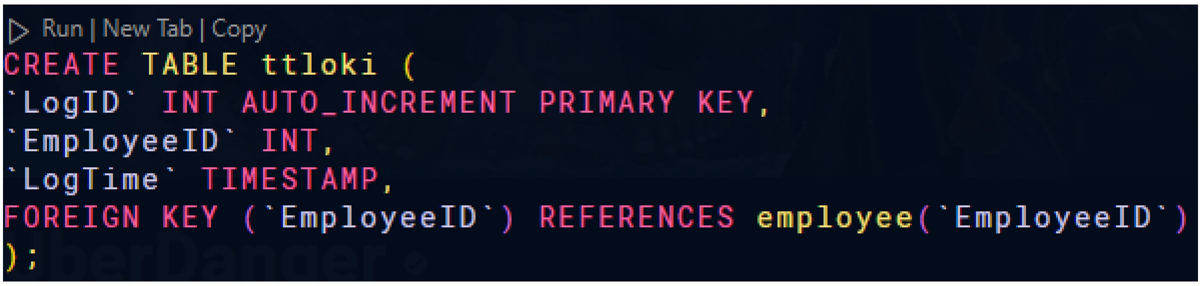
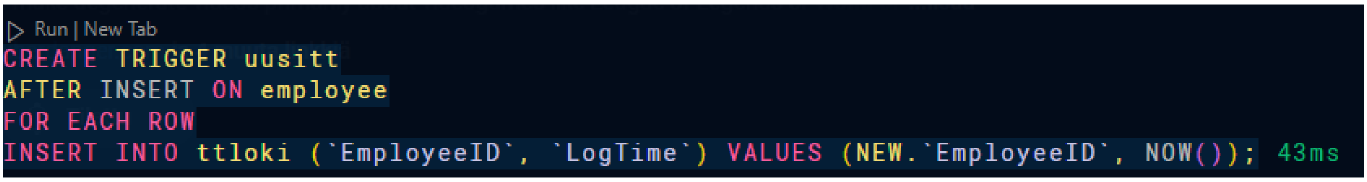
harj_liipaisimet_yms_v2.pdf

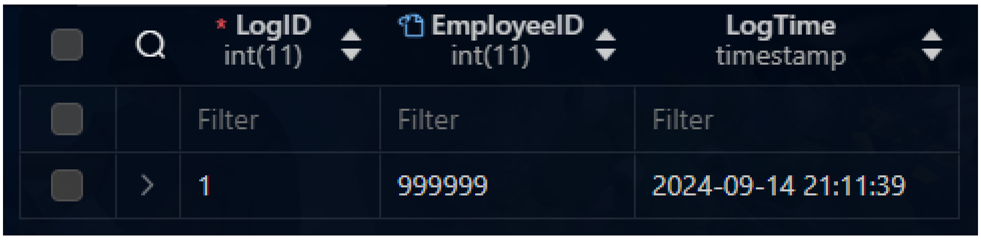
Task 1



```
CREATE TABLE ttloki (
`LogID` INT AUTO_INCREMENT PRIMARY KEY,
`EmployeeID` INT,
`LogTime` TIMESTAMP,
FOREIGN KEY (`EmployeeID`) REFERENCES employee(`EmployeeID`)
);
```



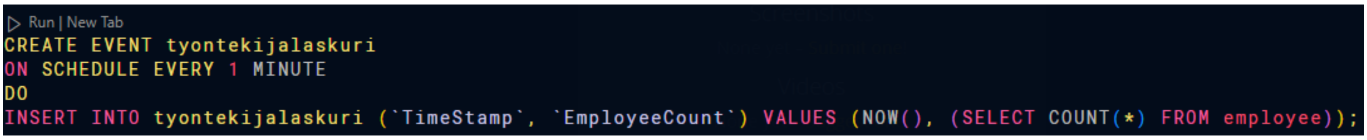
```
CREATE TRIGGER uusitt
AFTER INSERT ON employee
FOR EACH ROW
INSERT INTO ttloki (`EmployeeID`, `LogTime`) VALUES (NEW.`EmployeeID`, NOW());
```



LogID int(11)	EmployeeID int(11)	LogTime timestamp
Filter	Filter	Filter
1	999999	2024-09-14 21:11:39

The `SHOW CREATE TRIGGER` statement shows the name of the trigger, the trigger body, as well as some other information such as the time the trigger was created.

Task 2



```
CREATE EVENT tyontekijalaskuri
ON SCHEDULE EVERY 1 MINUTE
DO
INSERT INTO tyontekijalaskuri (`TimeStamp`, `EmployeeCount`) VALUES (NOW(), (SELECT COUNT(*) FROM employee));
```

The event triggers every minute, and logs the current timestamp and the number of employees.

Task 3

One row size:

- 4 bytes `int(11)`
- 4 bytes `timestamp`

Total: 8 bytes

1Gb = 1024Mb = 1024 * 1024 Kb = 1024 * 1024 * 1024 bytes

1024 * 1024 * 1024 / 8 = 134,217,728 rows

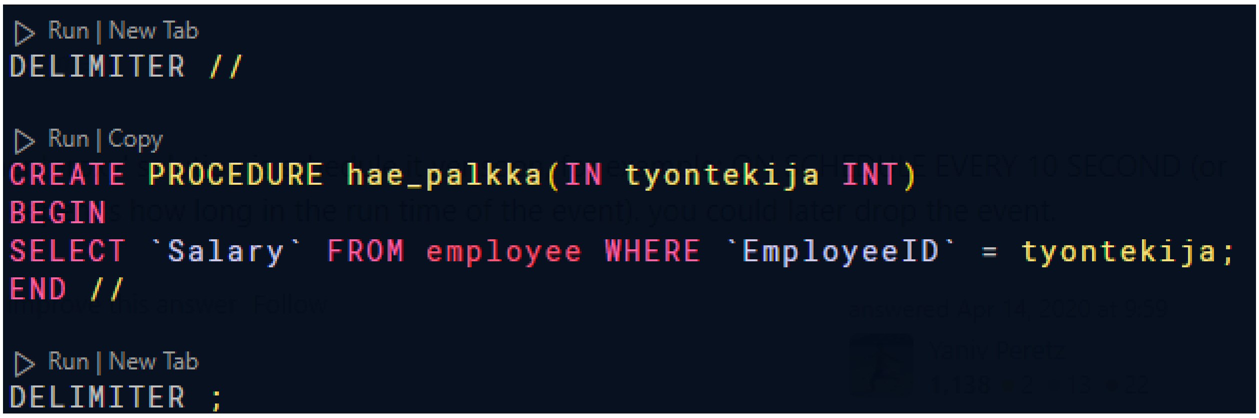
Row is created every minute, so it takes 134,217,728 minutes to get to 1Gb.

134,217,728 / 60 = 2,236,962 hours

2,236,962 / 24 = 93,206 days

93,206 / 365 = 255 years.

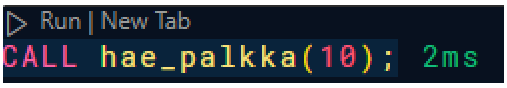
Task 4



```
DELIMITER //

CREATE PROCEDURE hae_palkka(IN tyontekija INT)
BEGIN
SELECT `Salary` FROM employee WHERE `EmployeeID` = tyontekija;
END //

DELIMITER ;
```



```
CALL hae_palkka(10);
```

When the procedure is called, it will select the salary of the employee with the given ID and return it. If the employee does not exist, it will return nothing.