UPDATE fine f, payment p

SET f.date\_payment = p.date\_payment,

f.sum\_fine = IF(((f.date\_payment - f.date\_violation) <= 20), f.sum\_fine/2, f.sum\_fine)

WHERE (f.name = p.name) AND (f.number\_plate = p.number\_plate) AND (f.violation = p.violation) AND (f.date\_payment is Null);

SELECT \* FROM fine;

SELECT author,

SUM(amount \* price) AS Стоимость

FROM book

WHERE title <> 'Идиот' or 'Белая гвардия'

GROUP BY author

HAVING SUM(amount \* price) > 5000

ORDER BY (SUM(amount \* price)) DESC;

SELECT author, title, amount

FROM book

WHERE amount IN(

SELECT amount

FROM book

GROUP BY amount

HAVING count(amount) = 1

);

SELECT title, author, amount, ((SELECT MAX(amount) FROM book) - amount) AS Заказ

FROM book

WHERE ABS(amount < (SELECT MAX(amount) FROM book));

INSERT INTO book (title, author, price, amount)

SELECT title, author, price, amount

FROM supply

WHERE author NOT IN (

SELECT author

FROM book

);

SELECT \* FROM book;

UPDATE book

SET buy = IF(buy > amount, amount, buy), price = IF(buy = 0, price \* 0.9, price);

SELECT \* FROM book;

UPDATE book, supply

SET book.amount = book.amount + supply.amount, book.price = IF(book.title = supply.title, (supply.price + book.price)/2, supply.price)

WHERE book.title = supply.title AND book.author = supply.author;

SELECT \* FROM book;

CREATE TABLE ordering AS

SELECT author, title,

(

SELECT ROUND(AVG(amount))

FROM book

) AS amount

FROM book

WHERE amount < (SELECT ROUND(AVG(amount))

FROM book

);

SELECT \* FROM ordering;

SELECT name, city, (DATEDIFF(date\_last, date\_first)+1) AS Длительность

FROM trip

WHERE city <> 'Москва' AND city <> 'Санкт-Петербург'

ORDER BY Длительность DESC, city DESC;

SELECT name, city, date\_first, ((DATEDIFF(date\_last, date\_first)+1)\*per\_diem) AS Сумма

FROM trip

WHERE (MONTHNAME(date\_first) IN ('February', 'March')) AND (YEAR(date\_first) = 2020)

ORDER BY name, Сумма DESC;

SELECT name, city, date\_first, date\_last

FROM trip

WHERE MONTH(date\_first) = MONTH(date\_last)

ORDER BY city, name;

SELECT MONTHNAME(date\_first) AS Месяц, COUNT(name) AS Количество

FROM trip

GROUP BY MONTHNAME(date\_first)

ORDER BY Количество DESC, Месяц;

UPDATE fine f, traffic\_violation tv

SET f.sum\_fine = tv.sum\_fine

WHERE (f.sum\_fine IS NULL) AND (f.violation = tv.violation);

SELECT name, number\_plate, violation

FROM fine

GROUP BY name, number\_plate, violation

HAVING count(\*) >= 2

ORDER BY name, number\_plate, violation;

CREATE TABLE query\_in (SELECT name, number\_plate, violation

FROM fine

GROUP BY name, number\_plate, violation

HAVING count(\*) >= 2

ORDER BY name, number\_plate, violation);

/\*SELECT \* FROM query\_in\*/

UPDATE fine f, query\_in q

SET f.sum\_fine = f.sum\_fine \* 2

WHERE (f.name = q.name) AND (f.number\_plate = q.number\_plate) AND (f.date\_payment is Null);

/\*GROUP BY number\_plate, violation

ORDER BY name, number\_plate, violation;\*/

SELECT \* FROM fine

UPDATE fine f, payment p

SET f.date\_payment = p.date\_payment,

f.sum\_fine = IF(((f.date\_payment - f.date\_violation) <= 20), f.sum\_fine/2, f.sum\_fine)

WHERE (f.name = p.name) AND (f.number\_plate = p.number\_plate) AND (f.violation = p.violation) AND (f.date\_payment is Null);

SELECT \* FROM fine;

CREATE TABLE back\_payment (

SELECT name, number\_plate, violation, sum\_fine, date\_violation

FROM fine

WHERE date\_payment is Null);

SELECT \* FROM back\_payment;

SELECT name\_city, name\_author, DATE\_ADD('2020-01-01', INTERVAL (FLOOR(RAND() \* 365)) DAY) AS Дата

FROM author CROSS JOIN city

ORDER BY name\_city, Дата DESC;

SELECT name\_author, SUM(amount) AS Количество

FROM

author LEFT JOIN book

on author.author\_id = book.author\_id

GROUP BY name\_author

HAVING (Количество < 10) OR (COUNT(title) = 0)

/\*(Количество IS NULL)\*/

ORDER BY Количество;

SELECT title, name\_author, name\_genre, price, amount

FROM

book

INNER JOIN author ON book.author\_id = author.author\_id

INNER JOIN genre ON book.genre\_id = genre.genre\_id

WHERE genre.genre\_id IN

(

SELECT top\_genre.genre\_id

FROM

(

SELECT genre\_id, SUM(amount) AS sum\_amount

FROM book

GROUP BY genre\_id

) top\_genre

INNER JOIN

(

SELECT genre\_id, SUM(amount) AS sum\_amount

FROM book

GROUP BY genre\_id

ORDER BY sum\_amount DESC

LIMIT 1

) max\_sum\_ammount\_on\_genre\_id

ON top\_genre.sum\_amount = max\_sum\_ammount\_on\_genre\_id.sum\_amount

)

ORDER BY title;

SELECT book.title AS Название, name\_author AS Автор, (supply.amount + book.amount) AS Количество

FROM

author

INNER JOIN book USING (author\_id)

INNER JOIN supply ON book.title = supply.title

and author.name\_author = supply.author

and book.price = supply.price;

UPDATE book

INNER JOIN author ON author.author\_id = book.author\_id

INNER JOIN supply ON book.title = supply.title

and supply.author = author.name\_author

SET book.amount = book.amount + supply.amount,

supply.amount = 0,

book.price = (book.price\*book.amount + supply.price\*supply.amount)/(book.amount+supply.amount)

WHERE book.price != supply.price;

SELECT \* FROM book;

SELECT \* FROM supply;

DELETE FROM author

WHERE author\_id IN (SELECT author\_id

FROM book

GROUP BY author\_id

HAVING (SUM(amount) <= 20));

SELECT \* FROM author;

SELECT \* FROM book;

DELETE FROM author

USING

author

INNER JOIN book ON author.author\_id = book.author\_id

INNER JOIN genre ON book.genre\_id = genre.genre\_id

WHERE genre.name\_genre = 'Поэзия';

SELECT \* FROM author;

SELECT \* FROM book;

SELECT DISTINCT author.name\_author, book.title, COUNT(buy\_book.amount) AS Количество

FROM

book

LEFT JOIN buy\_book ON book.book\_id = buy\_book.book\_id

JOIN author ON book.author\_id = author.author\_id

GROUP BY book.title, name\_author

ORDER BY author.name\_author, book.title;

SELECT buy\_id,

DATEDIFF(date\_step\_end,date\_step\_beg) AS Количество\_дней,IF

(DATEDIFF(date\_step\_end,date\_step\_beg) > days\_delivery,

DATEDIFF(date\_step\_end,date\_step\_beg) - days\_delivery,

0) AS Опоздание

FROM buy\_step INNER JOIN step USING (step\_id)

INNER JOIN buy USING (buy\_id)

INNER JOIN client USING (client\_id)

INNER JOIN city USING (city\_id)

where buy\_step.step\_id = 3 AND date\_step\_end IS NOT NULL

ORDER BY buy\_id;

SELECT genre.name\_genre, SUM(buy\_book.amount) AS Количество

FROM

buy\_book

JOIN book USING(book\_id)

JOIN genre USING(genre\_id)

GROUP BY genre.name\_genre

HAVING SUM(buy\_book.amount) = (

SELECT MAX(sum\_amount) AS max\_sum\_amount

FROM

(SELECT book.genre\_id, SUM(buy\_book.amount) AS sum\_amount

FROM genre

JOIN book USING(genre\_id)

JOIN buy\_book USING(book\_id)

GROUP BY book.genre\_id

) query\_in

);

SELECT YEAR(date\_payment) AS Год, MONTHNAME(date\_payment) AS Месяц, SUM(amount \* price) AS Сумма

FROM

(SELECT buy\_id, client\_id, book\_id, date\_payment, amount, price

FROM

buy\_archive

UNION ALL

SELECT buy.buy\_id, client\_id, book\_id, date\_step\_end, buy\_book.amount, price

FROM

book

INNER JOIN buy\_book USING(book\_id)

INNER JOIN buy USING(buy\_id)

INNER JOIN buy\_step USING(buy\_id)

INNER JOIN step USING(step\_id)

WHERE date\_step\_end IS NOT Null and name\_step = "Оплата"

ORDER BY client\_id) AS Uny

GROUP BY MONTHNAME(date\_payment), YEAR(date\_payment)

SELECT title, SUM(Uny.sum\_amount) AS Количество, SUM(Uny.sum\_price) AS Сумма

FROM

(SELECT title, SUM(buy\_archive.amount) AS sum\_amount, SUM(buy\_archive.amount \* buy\_archive.price) AS sum\_price

FROM

buy\_archive

JOIN book USING(book\_id)

GROUP BY title

UNION ALL

SELECT book.title, SUM(buy\_book.amount) as sum\_amount, SUM(buy\_book.amount \* book.price) as sum\_price

FROM

book

INNER JOIN buy\_book USING(book\_id)

INNER JOIN buy USING(buy\_id)

INNER JOIN buy\_step USING(buy\_id)

INNER JOIN step USING(step\_id)

WHERE step.name\_step = 'Оплата' AND buy\_step.date\_step\_end IS NOT NULL

GROUP BY title

) AS Uny

GROUP BY title

ORDER BY Сумма DESC;

CREATE TABLE buy\_pay(

SElECT title, name\_author, price, buy\_book.amount, SUM(price \* buy\_book.amount) AS Стоимость

FROM book

JOIN author USING(author\_id)

JOIN buy\_book USING(book\_id)

WHERE buy\_book.buy\_id = 5

GROUP BY title, name\_author, price, buy\_book.amount

ORDER BY title);

SELECT \* FROM buy\_pay;

UPDATE buy\_step

SET date\_step\_end = '2020-04-13'

WHERE buy\_id = 5 AND step\_id = (SELECT step\_id

FROM step

WHERE name\_step = 'Оплата');

UPDATE buy\_step

SET date\_step\_beg = '2020-04-13'

WHERE buy\_id = 5 AND step\_id = (SELECT step\_id + 1

FROM step

WHERE name\_step = 'Оплата');

SELECT \* FROM buy\_step;

SELECT name\_student, result

FROM attempt

JOIN student USING(student\_id)

GROUP BY name\_student, result

HAVING result =

(SELECT MAX(result) AS max\_result

FROM

(

SELECT result

FROM attempt

GROUP BY result) query\_in

)

ORDER BY name\_student;

SELECT question\_id, name\_question

FROM question

JOIN subject USING(subject\_id)

WHERE name\_subject = 'Основы баз данных'

GROUP BY question\_id, name\_question

ORDER BY RAND()

LIMIT 3;

SELECT name\_subject,

CONCAT(LEFT(name\_question, 30), '...') AS Вопрос,

COUNT(answer.answer\_id) AS Всего\_ответов,

ROUND(((SUM(answer.is\_correct) / COUNT(answer.is\_correct))\*100),2) AS Успешность

FROM subject

JOIN question on subject.subject\_id = question.subject\_id

JOIN answer on question.question\_id = answer.question\_id

JOIN testing on answer.answer\_id = testing.answer\_id

GROUP BY name\_subject, Вопрос

ORDER BY name\_subject, Успешность DESC, Вопрос;

INSERT INTO testing (attempt\_id, question\_id)

SELECT attempt\_id, question\_id

FROM attempt

JOIN question ON attempt.subject\_id = question.subject\_id

WHERE attempt\_id = (SELECT MAX(attempt\_id) FROM attempt)

ORDER BY RAND() LIMIT 3;

SELECT \* FROM testing;

UPDATE attempt

SET result = (

SELECT ROUND(((SUM(answer.is\_correct)) / 3 \* 100),0)

FROM testing

JOIN answer ON testing.answer\_id = answer.answer\_id

WHERE attempt\_id = 8)

WHERE attempt\_id = 8;

SELECT \* FROM attempt;

SELECT name\_subject, COUNT(enrollee\_subject.subject\_id) AS Количество, ROUND(MAX(result)) AS Максимум, ROUND(MIN(result)) AS Минимум, ROUND(AVG(result),1) AS Среднее

FROM enrollee\_subject

JOIN subject ON enrollee\_subject.subject\_id = subject.subject\_id

GROUP BY enrollee\_subject.subject\_id

ORDER BY name\_subject;

SELECT name\_enrollee, (IF(SUM(bonus) IS NULL, 0, SUM(bonus))) AS Бонус

FROM enrollee

LEFT JOIN enrollee\_achievement ON enrollee.enrollee\_id = enrollee\_achievement.enrollee\_id

LEFT JOIN achievement ON enrollee\_achievement.achievement\_id = achievement.achievement\_id

GROUP BY name\_enrollee

ORDER BY name\_enrollee;

SELECT name\_department, name\_program, plan, COUNT(enrollee\_id) AS Количество,

ROUND(((COUNT(enrollee\_id))/plan),2) AS Конкурс

FROM program\_enrollee

JOIN program ON program\_enrollee.program\_id = program.program\_id

JOIN department ON program.department\_id = department.department\_id

GROUP BY name\_department, name\_program, plan

ORDER BY Конкурс DESC;

SELECT name\_program

FROM program

JOIN program\_subject ON program.program\_id = program\_subject.program\_id

JOIN subject ON program\_subject.subject\_id = subject.subject\_id

WHERE name\_subject IN ('Информатика', 'Математика')

GROUP BY name\_program

HAVING COUNT(name\_subject) = 2

ORDER BY name\_program;

SELECT name\_program, name\_enrollee, SUM(result) AS itog

FROM enrollee

JOIN program\_enrollee ON enrollee.enrollee\_id = program\_enrollee.enrollee\_id

JOIN program ON program\_enrollee.program\_id = program.program\_id

JOIN program\_subject ON program.program\_id = program\_subject.program\_id

JOIN subject ON program\_subject.subject\_id = subject.subject\_id

JOIN enrollee\_subject ON subject.subject\_id = enrollee\_subject.subject\_id

and enrollee\_subject.enrollee\_id = enrollee.enrollee\_id

GROUP BY name\_program, name\_enrollee

ORDER BY name\_program, itog DESC;

SELECT name\_program, name\_enrollee

FROM enrollee

JOIN program\_enrollee ON enrollee.enrollee\_id = program\_enrollee.enrollee\_id

JOIN program ON program\_enrollee.program\_id = program.program\_id

JOIN program\_subject ON program.program\_id = program\_subject.program\_id

JOIN subject ON program\_subject.subject\_id = subject.subject\_id

JOIN enrollee\_subject ON subject.subject\_id = enrollee\_subject.subject\_id

and enrollee\_subject.enrollee\_id = enrollee.enrollee\_id

WHERE result < min\_result

GROUP BY name\_program, name\_enrollee

ORDER BY name\_program, name\_enrollee;

CREATE TABLE applicant AS (SELECT program.program\_id, enrollee.enrollee\_id, SUM(result) AS itog

FROM enrollee

JOIN program\_enrollee ON enrollee.enrollee\_id = program\_enrollee.enrollee\_id

JOIN program ON program\_enrollee.program\_id = program.program\_id

JOIN program\_subject ON program.program\_id = program\_subject.program\_id

JOIN subject ON program\_subject.subject\_id = subject.subject\_id

JOIN enrollee\_subject ON subject.subject\_id = enrollee\_subject.subject\_id

and enrollee\_subject.enrollee\_id = enrollee.enrollee\_id

GROUP BY program.program\_id, enrollee.enrollee\_id

ORDER BY program.program\_id, itog DESC);

SELECT \* FROM applicant;

DELETE FROM applicant

USING applicant

JOIN enrollee ON applicant.enrollee\_id = enrollee.enrollee\_id

JOIN program\_enrollee ON enrollee.enrollee\_id = program\_enrollee.enrollee\_id

JOIN program ON program\_enrollee.program\_id = program.program\_id

JOIN program\_subject ON program.program\_id = program\_subject.program\_id

JOIN subject ON program\_subject.subject\_id = subject.subject\_id

JOIN enrollee\_subject ON subject.subject\_id = enrollee\_subject.subject\_id

and enrollee\_subject.enrollee\_id = enrollee.enrollee\_id

WHERE result < min\_result AND applicant.program\_id = program.program\_id;

SELECT \* FROM applicant;