**Задача 1**

* Показать все названия книг вместе с именами издателей.

SELECT TITLE, PUBNAME

FROM BOOK

* В какой книге наибольшее количество страниц?

SELECT \* FROM BOOK

WHERE PAGESNUM = MAX(PAGESNUM);

* Какие авторы написали более 5 книг?

SELECT AUTHOR

FROM BOOK

GROUP BY AUTHOR HAVING COUNT(\*) > 5;

* В каких книгах более чем в два раза больше страниц, чем среднее количество страниц для всех книг?

SELECT \* FROM BOOK

WHERE PAGESNUM > (SELECT AVG(PAGESNUM) \* 2 FROM BOOK);

* Какие категории содержат подкатегории?

SELECT DISTINCT a. CATEGORYNAME

FROM CATEGORY a INNER JOIN CATEGORY b ON a. CATEGORYNAME = b.PARENTCAT;

* У какого автора (предположим, что имена авторов уникальны) написано максимальное количество книг?
* Какие читатели забронировали все книги (не копии), написанные "Марком Твеном"?
* Какие книги имеют более одной копии?

SELECT ISBN, COUNT(\*) as BOOKCOUNT FROM Copy

GROUP BY ISBN HAVING BOOKCOUNT > 1;

* ТОП 10 самых старых книг

SELECT \* FROM BOOK ORDER BY PUBYEAR LIMIT 10;

* Перечислите все категории в категории “Спорт” (с любым уровнем вложености).

**Задача 2**

Напишите SQL-запросы для следующих действий:

* Добавьте запись о бронировании читателем ‘Василеем Петровым’ книги с ISBN 123456 и номером копии 4.

INSERT INTO BORROWING VALUES (‘Василеем Петровым’ , 123456 ,4, NULL);

* Удалить все книги, год публикации которых превышает 2000 год.

DELETE FROM BOOK WHERE PubYear > 2000;

DELETE FROM COPY

WHERE ISBN IN (SELECT ISBN FROM BOOK WHERE PubYear > 2000);

DELETE FROM BORROWING

WHERE ISBN IN (SELECT ISBN FROM BOOK WHERE PubYear > 2000);

DELETE FROM BOOKCAT

WHERE ISBN IN (SELECT ISBN FROM BOOK WHERE PubYear > 2000);

* Измените дату возврата для всех книг категории "Базы данных", начиная с 01.01.2016, чтобы они были в заимствовании на 30 дней дольше (предположим, что в SQL можно добавлять числа к датам).

UPDATE BORROWING

SET RETURNDATE = RETURNDATE + 30

WHERE RETURNDATE >= DATE("01.01.2016") AND ISBN IN

(SELECT ISBN FROM BOOKCAT WHERE CATEGORYNAME = "Базы данных");

**Задача 3**

Рассмотрим следующую реляционную схему:

* Student( MatrNr, Name, Semester )
* Check( MatrNr, LectNr, ProfNr, Note )
* Lecture( LectNr, Title, Credit, ProfNr )
* Professor( ProfNr, Name, Room )

Опишите на русском языке результаты следующих запросов:

SELECT s.Name, s.MatrNr FROM Student s

WHERE NOT EXISTS (

SELECT \* FROM Check c WHERE c.MatrNr = s.MatrNr AND c.Note >= 4.0 ) ;

Возвращается имя и номер студентов с оценками меньше 4.0

( SELECT p.ProfNr, p.Name, sum(lec.Credit)

FROM Professor p, Lecture lec

WHERE p.ProfNr = lec.ProfNr

GROUP BY p.ProfNr, p.Name)

UNION

( SELECT p.ProfNr, p.Name, 0

FROM Professor p

WHERE NOT EXISTS (

SELECT \* FROM Lecture lec WHERE lec.ProfNr = p.ProfNr ));

Возвращается номер, имя и сумма всех кредитов у профессоров. Если у профессора нет лекций, он получает 0 кредитов.

SELECT s.Name, p.Note

FROM Student s, Lecture lec, Check c

WHERE s.MatrNr = c.MatrNr AND lec.LectNr = c.LectNr AND c.Note >= 4

AND c.Note >= ALL (

SELECT c1.Note FROM Check c1 WHERE c1.MatrNr = c.MatrNr )

Возвращается имя и оценка студентов, имеющих наивысший балл больше или равный 4