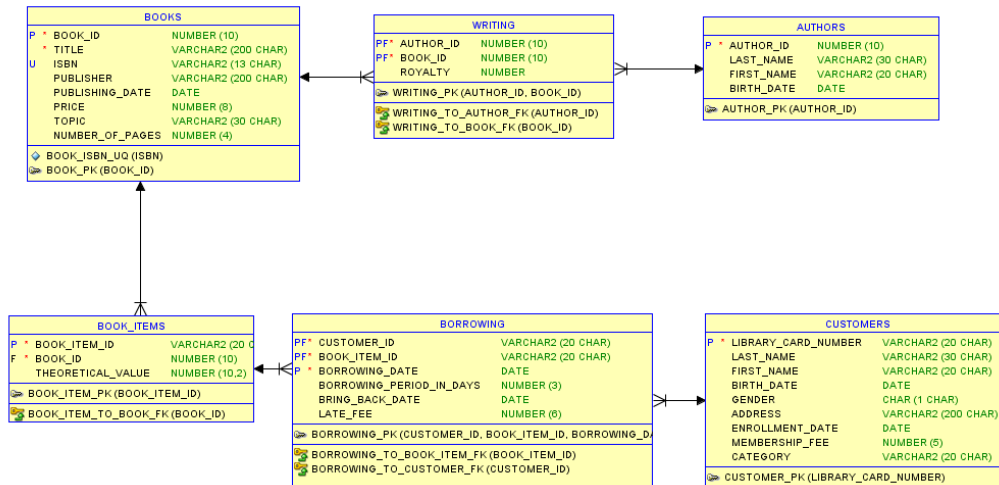


# SQL tasks and their solutions for BOOK\_LIBRARY schema for Computer Science BSc students

Made by Vágner Anikó

2018

This work was supported by the construction EFOP-3.4.3-16-2016-00021.  
The project was supported by the European Union, co-financed by the European Social Fund.



## 1 Select statements with order by clause

**Example 1** List the names and birth date of the authors. The list should be sorted by name descending and birth date ascending.

```

select last_name || ' ' || first_name name,
       to_char(BIRTH_DATE, 'yyyy.month.dd_hh24:mi:ss') bdate
from book_library.authors
  
```

```
order by name desc, bdate asc;
```

**Example 2** *List the books. Sort the list by number of pages descending. Put the rows to the end of the list where the number of pages is not given.*

```
select *  
from book_library.books  
order by number_of_pages desc nulls last;
```

**Example 3** *List the books. Sort the list by number of pages ascending. Put the rows to the beginning of the list where the number of pages is not given.*

```
select *  
from book_library.books  
order by number_of_pages nulls first;
```

## 2 Select statements with where clause

**Example 4** *List the title, price, number of pages, topic and publisher of the books which price is more than 1000 and which number of pages is less than 100. Sort the list by price descending.*

```
select title, price, number_of_pages, topic, publisher  
from book_library.books  
where price>1000 and number_of_pages<100  
order by price desc;
```

**Example 5** *Find the books which price is less than 4000 and which topic is Fairy Tales or which price is more than 2000 and which topic is Thriller. Sort the result by topic descending and by number of pages.*

```
select title, price, topic, number_of_pages  
from book_library.books  
where price<4000 and topic='Fairy_Tales'  
      or price>2000 and topic='Thriller'  
order by topic desc, number_of_pages
```

**Example 6** *List the books which topic is Fairy Tales or which price is more than 3000. Sort the list by price.*

```
select title, topic, price
from book_library.books
where topic='Fairy_Tales' or price>3000
order by price
```

**Example 7** *List the books which price is between 3000 and 4000.*

- ```
select title, price
from book_library.books
where price >=3000 and price <=4000;
```
- ```
select title, price
from book_library.books
where price between 3000 and 4000;
```

**Example 8** *List the books which price is not between 3000 and 4000.*

```
select title, price
from book_library.books
where price not between 3000 and 4000;
```

**Example 9** *List the books which topic is Fairy Tales, Thriller or History.*

- ```
select title, topic
from book_library.books
where topic='Fairy_Tales' or topic='Thriller' or topic='History';
```
- ```
select title, topic
from book_library.books
where topic in ('Fairy_Tales','Thriller', 'History');
```

**Example 10** *List the books which topic is not Fairy Tales, not Thriller and not History.*

```
select title, topic
from book_library.books
where topic not in ('Fairy_Tales','Thriller', 'History');
```

**Example 11** *List the books which topic is given.*

```
select *
from book_library.books
where topic is null;
```

**Example 12** *List the books which topic is not given.*

```
select *
from book_library.books
where topic is not null;
```

**Example 13** *List the books which topic is Science Fiction or Literature or which price is between 1000 and 4000 and which number of pages is given. Sort the list by topic descending and number of pages ascending.*

```
select *
from book_library.books
where (topic in ('Science_Fiction','Literature')
      or price between 1000 and 4000)
      and number_of_pages is not null
order by topic desc, number_of_pages;
```

**Example 14** *List the female customers. Sort the list by last name descending and first name ascending.*

```
select *
from book_library.customers
where gender='f'
order by last_name desc, first_name asc;
```

**Example 15** *List the female pensioner customers. Sort the list by last name descending and first name ascending.*

```
select *
from book_library.customers
where gender='f' and category='pensioner'
order by last_name desc, first_name asc;
```

**Example 16** *List the female or pensioner customers. Sort the list by gender descending.*

```
select *
from book_library.customers
where gender='f' or category='pensioner'
order by gender desc;
```

**Example 17** *List the customers who are not female or pensioner. Sort the list by gender descending.*

```
select *
from book_library.customers
where not (gender='f' or category='pensioner')
order by gender desc;
```

**Example 18** *List the customers who are pensioner, student or child. Sort the list by gender descending.*

- ```
select *
from book_library.customers
where category='pensioner' or category='student' or category='child'
order by gender desc;
```
- ```
select *
from book_library.customers
where category in ('pensioner','student','child')
order by gender desc;
```

**Example 19** *List the customers who are not pensioner nor student nor child. Sort the list by gender descending.*

```
select *
from book_library.customers
where category not in ('pensioner','student','child')
order by gender desc;
```

**Example 20** *List the books which title contains at least one 'n' letter.*

```
select *
from book_library.books
where title like '%n%';
```

**Example 21** *List the books which title's second letter is 'n'.*

```
select *
from book_library.books
where title like '_n%';
```

**Example 22** *List the books which title's second letter is not 'n'.*

```
select *
from book_library.books
where title not like '_n%';
```

**Example 23** *Write comment.*

- `--where title not like '_n%';`
- `/*alkfglkj  
asdjkhf*/`

**Example 24** *List the books which topic is thriller and which number of pages is less than 400. Sort the list by the number of pages.*

```
select *  
from book_library.books  
where topic='Thriller'  
and number_of_pages<400  
order by number_of_pages;
```

**Example 25** *List the books where it is not true that which topic is thriller and which number of pages is less than 400. Sort the list by the number of pages.*

```
select *  
from book_library.books  
where not (topic='Thriller'  
and number_of_pages<400)  
order by number_of_pages;
```

**Example 26** *List the books which title contains at least one '0' character.*

```
select *  
from book_library.books  
where title like '%0%';
```

**Example 27** *List the books which title's third character is 'a'.*

```
select *  
from book_library.books  
where title like '__a%';
```

**Example 28** *List the books which title's third character is not 'a'.*

```
select *  
from book_library.books  
where title not like '__a%';
```

**Example 29** *What is theoretical value and the book\_item\_id of the book\_items whose book\_id is 1116152201? Sort the list by the theoretical value descending.*

```
select theoretical_value, book_item_id
from book_library.book_items
where book_id='1116152201'
order by theoretical_value desc;
```

**Example 30** *List the title of the books which number of pages is between 150 and 200 or whose publisher is neither Arrow nor Signet and nor Apress? Sort the list.*

```
select title
from book_library.books
where number_of_pages between 150 and 200
or publisher not in ('Arrow','Signet','Apress')
order by title;
```

### 3 Functions and expressions in select statements

**Example 31** *List the names of customers.*

- ```
select last_name, first_name
from book_library.customers;
```
- ```
select last_name||' '|| first_name as full_name
from book_library.customers;
```

**Example 32** *List the names of customers. Sort the list by name.*

```
select last_name||' '|| first_name full_name
from book_library.customers
order by last_name, first_name;
```

**Example 33** *List the names of customers. Sort the list by last name descending and first name ascending.*

```
select last_name||' '|| first_name full_name
from book_library.customers
order by last_name desc, first_name asc;
```

**Example 34** *List the title of the books. Put the 1 value into the second. Put the 'a' letter into the third column and into the last column put the current date and time.*

```
select title, 1, 'a', to_char(sysdate, 'yyyy.mm.dd_hh24:mi:ss')
from book_library.books;
```

**Example 35** *List the price per page for each book. Try out the trunc and round functions with these values.*

```
select price/number_of_pages, trunc(price/number_of_pages),
       trunc(price/number_of_pages,2),
       round(price/number_of_pages), round(price/number_of_pages,2)
from book_library.books;
```

**Example 36** *List your username.*

```
select user
from dual;
```

**Example 37** *List the concatenation of the 'apple', 'tree' and 'banana' words.*

- ```
select 'apple' || 'tree' || 'banana'
from dual;
```
- ```
select concat(concat('apple','tree'),'banana')
from dual;
```

**Example 38** *List the address of the customers. In the first column as it is stored in the database, in the second column only with lowercase letters and in the third column only with uppercase letters.*

```
select address, lower(address), upper(address)
from book_library.customers;
```

**Example 39** *List the address of customers, and the substring of the address from the 3rd character to the 7th character.*

```
select address,substr(address,3,5)
from book_library.customers;
```

**Example 40** *List the address of customers, and in the second column replace the 'Str' string in the address to the '\*\*\*\*\*'.*

```
select address,replace(address,'Str','*****')
from book_library.customers;
```

**Example 41** *List the address of customers, and in the second column the number of the character from which the 'Str' string starts.*

```
select address,instr(address,'Str')
from book_library.customers;
```

**Example 42** *List the address of customers, and in the second column the number of the character in the address.*

```
select address,length(address)
from book_library.customers;
```

**Example 43** *List your username and the current date and time.*

```
select user, sysdate
from dual;
```

**Example 44** *List the topics of the books. In the second column list again the topics, but if the topic is null list '\*\*\*\*\*' instead of it.*

```
select topic, nvl(topic,'*****')
from book_library.books;
```

**Example 45** *List the gender of the customers. In the second column write 'male' if the gender is 'm', write 'female' if the gender is 'f', and write '?' otherwise.*

```
select gender, decode(gender, 'm','male','f','female','?')
from book_library.customers;
```

**Example 46** *List the birth date of the customers.*

```
select birth_date, to_char(BIRTH_DATE, 'yyyy.mm.dd_hh24:mi:ss')
from book_library.customers;
```

**Example 47** *List the name and birth date of the customers who was born earlier than 1st January, 2000.*

```
select last_name, first_name, to_char(BIRTH_DATE, 'yyyy.mon.dd_hh24:mi:ss')
from book_library.customers
where birth_date < to_date('2000.01.01', 'yyyy.mm.dd');
```

**Example 48** *List the date which is 2 months later than the current date.*

```
select to_char(add_months(sysdate, 2), 'yyyy.mon.dd')
from dual;
```

**Example 49** *How many years did pass away from 1st January, 2000?*

```
select months_between(sysdate, to_date('2000.01.01', 'yyyy.mm.dd'))/12
from dual;
```

**Example 50** *List the date which is 3 years later than the current date.*

```
select to_char(add_months(sysdate, 3*12), 'yyyy.mm.dd_hh24:mi:ss')
from dual;
```

**Example 51** *List the date which is 3 days later than the current date.*

```
select to_char(sysdate+3, 'yyyy.mm.dd_hh24:mi:ss')
from dual;
```

**Example 52** *List the date which is 3 hours later than the current date and time.*

```
select to_char(sysdate+3/24, 'yyyy.mm.dd_hh24:mi:ss')
from dual;
```

**Example 53** *List the name and birth date of customers who are less than 30 years old. Sort the list by the birth date.*

- ```
select last_name, first_name, to_char(birth_date,'yyyy.mm.dd') b_date
from book_library.customers
where birth_date<add_months(sysdate,-30*12)
order by b_date;
```
- ```
select last_name, first_name, to_char(birth_date,'yyyy.mm.dd') b_date
from book_library.customers
where months_between(sysdate,birth_date)>30*12
order by b_date;
```

**Example 54** *List the title, the book id and the publishing date of the books which has no price. Sort the list by publishing date descending and the title.*

```
select title, book_id, to_char(publishing_date,'yyyy.month_dd') pd
from book_library.books
where price is null
order by publishing_date desc, title;
```

**Example 55** *List the author id, name, birth date and age of the authors who was born before 2 January, 1990. Sort the list by age in reverse order.*

```
select author_id, first_name||' '||last_name name, to_char(birth_date,'yyyy.mm.dd'),
       months_between(sysdate,birth_date)/12 age
from book_library.authors
where birth_date<to_date('1990.01.02','yyyy.mm.dd')
order by age desc;
```

**Example 56** *What is the name of the author whose name contains punctually 2 'e'?*

```
select last_name||' '||first_name
from book_library.authors
where lower(last_name||first_name) like '%e%e%'
and lower(last_name||first_name) like '%e%e%e%';
```

## 4 Select statement with grouping functions, group by and having clause

**Example 57** *What is the maximum price, minimum price, the sum of the prices, the average of the prices of the books? How many books are in the book table?*

```
select max(price), min(price), sum(price), avg(price),
       count(price), count(*), count(book_id)
from book_library.books;
```

**Example 58** *How many books have been published by Arrow publisher?*

```
select count(*)
from book_library.books
where publisher='Arrow';
```

**Example 59** *What is the minimum price of the books which topic is Thriller?*

```
select min(price)
from book_library.books
where topic='Thriller';
```

**Example 60** *How many books belongs to each publisher?*

```
select publisher, count(*)
from book_library.books
group by publisher
order by publisher;
```

**Example 61** *What are the minimum price, the maximum price, the sum of the prices and the average of the prices of books for each publisher?*

```
select publisher, min(price), max(price), sum(price), avg(price)
from book_library.books
group by publisher
order by publisher;
```

**Example 62** *Which publisher did publish more than 3 books?*

```
select publisher
from book_library.books
group by publisher
having count(*)>3
order by publisher;
```

**Example 63** *Which are the publishers the sum of the prices of the books published by which is more than 20000? Sort the list.*

```
select publisher
from book_library.books
group by publisher
having sum(price)>20000
order by publisher;
```

**Example 64** *What is the sum of theoretical value of the book\_items for each book\_id?*

```
select book_id, sum(theoretical_value)
from book_library.book_items
group by book_id;
```

**Example 65** *What is the last borrowing date for each customer\_id in the cases when the customer brought back the book? Sort the list by date.*

```
select customer_id, to_char(max(borrowing_date), 'yyyy-mm-dd') mdb
from book_library.borrowing
where bring_back_date is not null
group by customer_id
order by mdb;
```

**Example 66** *List the topics which have less than 5 books.*

```
select topic
from book_library.books
group by topic
having count(*)<5;
```

**Example 67** *List the categories of the customers where the average membership fee is less than 2000.*

```
select category
from book_library.customers
group by category
having avg(membership_fee)<2000;
```

**Example 68** *How many customers belongs to each categories?*

```
select category, count(*)
from book_library.customers
group by category;
```

**Example 69** *What are id-s of the books the sum of theoretical values of the book\_items belonging to which is less than 5000?*

```
select book_id
from book_library.book_items
group by book_id
having sum(theoretical_value)<5000;
```

**Example 70** *Which books (only book\_id) have more than 4 instance (book\_items)?*

```
select book_id
from book_library.book_items
group by book_id
having count(*)>4;
```

**Example 71** *How many customers belong to each category?*

```
select category
from book_library.customers
group by category;
```

**Example 72** *How much is the maximum number of pages for each publisher?*

```
select publisher, max(number_of_pages)
from book_library.books
group by publisher;
```

## 5 Select statements with joins

**Example 73** *List the id-s of the book items and the titles and id-s of the books belonging to the book items.*

- ```
select bo.book_id, title, bit.book_item_id, bit.book_id
from book_library.books bo inner join book_library.book_items bit
on bo.book_id=bit.book_id;
```
- ```
select book_id, title, bit.book_item_id
from book_library.books bo inner join book_library.book_items bit
using (book_id);
```

- ```
select bo.book_id, title, bit.book_item_id, bit.book_id
from book_library.books bo, book_library.book_items bit
where bo.book_id=bit.book_id;
```
- ```
select book_id, title, bit.book_item_id
from book_library.books bo natural join book_library.book_items bit;
```

**Example 74** *What books (only book\_id) did Christie Agatha write?*

```
select book_id
from book_library.authors auth inner join book_library.writing w
  on auth.author_id=w.author_id
where last_name='Christie'
  and first_name='Agatha';
```

**Example 75** *Who is the author (only author\_id) of the book which title is 'Deathworld'?*

```
select w.author_id
from book_library.books bo inner join book_library.writing w
  on bo.book_id=w.book_id
where title='Deathworld_';
```

**Example 76** *What book\_item\_ids was borrowed by female customers?*

```
select book_item_id
from book_library.customers cu inner join book_library.borrowing bo
  on cu.library_card_number=bo.customer_id
where gender='f';
```

**Example 77** *Who is author (name of the author) of the book which title is 'Deathworld'?*

```
select last_name||'_'|| first_name
from book_library.books bo inner join BOOK_LIBRARY.WRITING w
  on bo.book_id=w.book_id
  inner join book_library.authors au
  on w.author_id=au.author_id
where title='Deathworld_';
```

**Example 78** *How many book items was borrowed by each customer (name)?*

```

select cu.LIBRARY_CARD_NUMBER, count(bo.BOOK_ITEM_ID), cu.first_name, cu.last_name
from book_library.customers cu inner join
    BOOK_LIBRARY.BORROWING bo
    on cu.LIBRARY_CARD_NUMBER=bo.CUSTOMER_ID
group by cu.LIBRARY_CARD_NUMBER, cu.first_name, cu.last_name

```

**Example 79** *How many authors did write the books which topic is history, science, or literature. One author should be counted only one time.*

```

select count(distinct w.author_id)
from book_library.writing w inner join
    BOOK_LIBRARY.BOOKS bo
    on w.book_id=bo.book_id
where topic in ('History', 'Science', 'Literature');

```

**Example 80** *List the titles of the books and the name of their authors authors of which were born earlier than 1950.*

```

select bo.title, au.first_name, au.last_name
from BOOK_LIBRARY.BOOKS bo inner join
    BOOK_LIBRARY.WRITING w
    on bo.book_id=w.book_id
    inner join book_library.authors au
    on w.author_id=au.author_id
where au.birth_date<to_date('1950','yyyy');

```

**Example 81** *How many authors did write books in each topic? One author should be counted only one time. Sort the list by the number of authors and topic.*

```

select bo.topic, count(distinct author_id) nu
from BOOK_LIBRARY.BOOKS bo inner join
    BOOK_LIBRARY.WRITING w
    on bo.book_id=w.book_id
group by bo.topic
order by nu, topic;

```

**Example 82** *How many authors are there in the author table whose birth date is not given?*

```

select count(*)
from book_library.authors
where birth_date is null;

```

**Example 83** *How many customers are there in the table for each gender who live in Debrecen city?*

```
select decode(gender,'m','male','f','female'), count(*)
from book_library.customers
where substr(address, instr(address,',')+2,
        length(address)-instr(address,',')-1)='Debrecen'
group by gender;
```

**Example 84** *How many customers live on addresses which contain the 'str' string (case-insensitive)?*

```
select count(*)
from book_library.customers
where lower(address) like '%str%';
```

**Example 85** *What are categories of the customers who borrowed more than 5 book items?*

```
select cu.category, count(book_item_id)
from book_library.customers cu inner join
        book_library.borrowing bo
        on cu.library_card_number=bo.customer_id
group by cu.category
having count(book_item_id)>5;
```

**Example 86** *How many books were written by each authors? List only the authors who wrote some books.*

```
select au.author_id, last_name, first_name, count(*)
from BOOK_LIBRARY.WRITING w inner join
        BOOK_LIBRARY.AUTHORS au
        on w.author_id=au.author_id
group by au.author_id, last_name, first_name;
```

**Example 87** *How many books were written by each authors? List all the authors (even if they did not write any books).*

- ```
select au.author_id, last_name, first_name, count(w.book_id)
from BOOK_LIBRARY.WRITING w right outer join
        BOOK_LIBRARY.AUTHORS au
        on w.author_id=au.author_id
group by au.author_id, last_name, first_name;
```

- ```
select au.author_id, last_name, first_name, count(w.book_id)
from BOOK_LIBRARY.WRITING w ,BOOK_LIBRARY.AUTHORS au
where w.author_id (+)=au.author_id
group by au.author_id, last_name, first_name;
```

**Example 88** *List all customers and the book\_item\_id-s which they borrowed. List also the customers who did not borrow anything. Order the list by name.*

- ```
select first_name, last_name, cu.library_card_number,bo.BOOK_ITEM_ID
from book_library.customers cu left outer join BOOK_LIBRARY.BORROWING bo
on cu.library_card_number=bo.customer_id
order by first_name, last_name;
```
- ```
select first_name, last_name, cu.library_card_number,bo.BOOK_ITEM_ID
from book_library.customers cu,BOOK_LIBRARY.BORROWING bo
where cu.library_card_number=bo.customer_id (+)
order by first_name, last_name;
```

**Example 89** *List all books (with id and title) and if they have book\_items put their book\_item\_id-s next to the title.*

- ```
select bo.book_id, title, it.book_item_id
from book_library.books bo left outer join book_library.book_items it
on bo.book_id=it.book_id;
```
- ```
select bo.book_id, title, it.book_item_id
from book_library.books bo, book_library.book_items it
where bo.book_id=it.book_id (+);
```

**Example 90** *How much is the sum of the late\_fee-s for each customers?*

```
select first_name, last_name, cu.LIBRARY_CARD_NUMBER, sum(nvl(late_fee,0))
from book_library.customers cu left outer join book_library.borrowing bo
on cu.LIBRARY_CARD_NUMBER=bo.CUSTOMER_ID
group by first_name, last_name, cu.LIBRARY_CARD_NUMBER;
```

**Example 91** *List the name, birth\_date, age of all authors. Sort the list by age descending and by name.*

```
select first_name, last_name,
to_char(birth_date,'yyyy.mm.dd'),
months_between(sysdate,birth_date)/12 age
from book_library.authors
order by age desc, first_name, last_name;
```

**Example 92** *List the title of the books which title contain at least two 'e' characters (case-insensitive) or which second letter is 'o'. Sort the list by title.*

```
select title from book_library.books
where lower(title) like '%e%e%' or
      lower(title) like '_o%'
order by title;
```

**Example 93** *How much is the average of the number of pages for each topic? Sort the list by topic.*

```
select topic, avg(number_of_pages)
from book_library.books
group by topic order by topic;
```

**Example 94** *Which category has more than 3 customers?*

```
select category
from BOOK_LIBRARY.CUSTOMERS
group by category
having count(*)>3;
```

**Example 95** *Who is the writer of the book which title is 'Napoleon'?*

```
select first_name, last_name
from book_library.authors au inner join
      book_library.writing w
      on au.author_id=w.author_id
      inner join
      book_library.books bo
      on w.book_id=bo.book_id
where title='Napoleon'
```

**Example 96** *List the name, birth\_date and category of all customers whose category is student, adult or supporter and the book\_item\_ids which they borrowed if they did any. Sort the list by the name of the customers and by the book\_item\_id.*

```

select first_name, last_name,
       to_char(birth_date), category,
       book_item_id
from book_library.customers cu left outer join
     book_library.borrowing bo
     on cu.library_card_number=bo.CUSTOMER_ID
where category in ('student', 'adult','supporter')
order by first_name, last_name, book_item_id;

```

**Example 97** *Which books have more than 3 writers?*

```

select title, bo.book_id
from book_library.books bo inner join BOOK_LIBRARY.WRITING w on bo.book_id=w.book_id
group by title, bo.book_id
having count(author_id)>3;

```

**Example 98** *Which books have less than 3 items?*

```

select title, bo.book_id, count(book_item_id)
from book_library.books bo left outer join BOOK_LIBRARY.BOOK_ITEMS bi
     on bo.book_id=bi.book_id
group by title, bo.book_id
having count(book_item_id)<3;

```

**Example 99** *List the customer whose is more than 20 years and who borrows the book (and they did not bring back) which theoretical value is not null?*

```

select first_name, last_name, cu.LIBRARY_CARD_NUMBER
from book_library.customers cu inner join book_library.borrowing bo
     on cu.LIBRARY_CARD_NUMBER=bo.CUSTOMER_ID
     inner join BOOK_LIBRARY.BOOK_ITEMS bi on bo.BOOK_ITEM_ID=bi.BOOK_ITEM_ID
where months_between(sysdate,cu.BIRTH_DATE)/12>20
     and bring_back_date is null
     and theoretical_value is not null;

```

**Example 100** *What is the maximum book price for each authors? List all authors, sort the list by the maximum price.*

```

select first_name, last_name, au.author_id, max(bo.PRICE) mp
from book_library.authors au left outer join BOOK_LIBRARY.WRITING w
     on au.author_id=w.author_id left outer join book_library.books bo
     on w.book_id=bo.book_id
group by first_name, last_name, au.author_id
order by mp;

```

**Example 101** *Which books has less than 3 items?*

```
select bo.book_id, title, count(book_item_id)
from book_library.books bo left outer join
book_library.book_items bit
on bo.book_id=bit.book_id
group by bo.book_id, title
having count(book_item_id)<3;
```

## 6 Select statements with subqueries

**Example 102** *Who is oldest customer?*

```
select *
from book_library.customers
where birth_date=(select min (birth_date)
                  from book_library.customers);
```

**Example 103** *Who are customers who did not borrowed anything?*

- ```
select *
from BOOK_LIBRARY.CUSTOMERS cu
where cu.LIBRARY_CARD_NUMBER not in (select customer_id
                                     from book_library.borrowing);
```
- ```
select cu.*
from BOOK_LIBRARY.CUSTOMERS cu left outer join book_library.borrowing bo
on cu.LIBRARY_CARD_NUMBER=bo.CUSTOMER_ID
where bo.BOOK_ITEM_ID is null;
```

**Example 104** *Who is the author (author\_id) of the book which title is Napoleon?*

- ```
select author_id
from book_library.books bo inner join BOOK_LIBRARY.WRITING w
on bo.book_id=w.book_id
where title='Napoleon';
```
- ```
select author_id
from BOOK_LIBRARY.WRITING
where book_id in (select book_id
                  from book_library.books
                  where title='Napoleon');
```

- ```
select author_id
from BOOK_LIBRARY.WRITING w inner join (select book_id
                                         from book_library.books
                                         where title='Napoleon') bo
on w.book_id=bo.book_id;
```

**Example 105** *List the author who got the highest royalty for a writing.*

```
select first_name, last_name author_id
from book_library.authors
where author_id in (select author_id
                    from book_library.writing
                    where royalty=(select max(royalty)
                                   from book_library.writing))
```

## 7 Select statements with set operators

**Example 106** *List the names of the authors and customers in one list. Every name should be listed only one time.*

```
select first_name, last_name
from book_library.authors
union
select first_name, last_name
from book_library.customers;
```

**Example 107** *List the names of the authors and customers in one list. If a name has more than on occasion, list it more times.*

```
select first_name, last_name
from book_library.authors
union all
select first_name, last_name
from book_library.customers;
```

**Example 108** *List the authors who are also customers.*

- ```
select first_name, last_name
from book_library.authors
intersect
select first_name, last_name
from book_library.customers;
```

- ```
select au.first_name, au. last_name
from book_library.authors au inner join book_library.customers cu
on au.first_name=cu.first_name and au.last_name=cu.last_name;
```

**Example 109** *List the names of the authors who are not customers.*

```
select first_name, last_name
from book_library.authors
minus
select first_name, last_name
from book_library.customers;
```

**Example 110** *List the authors who did not write any books.*

- ```
select first_name, last_name
from book_library.authors
where author_id not in (select author_id
from book_library.writing);
```
- ```
select first_name, last_name
from book_library.authors
where author_id in (select author_id
from book_library.authors
minus
select author_id
from book_library.writing);
```
- ```
select first_name, last_name
from book_library.authors au left outer join book_library.writing w
on au.author_id=w.author_id
where w.author_id is null;
```

## 8 Select statements using ROWNUM

**Example 111** *List the 5 oldest customers.*

```
select *
from (select *
from book_library.customers
order by birth_date)
where rownum<6
```

**Example 112** *List the first 5 customer's name in alphabetical order.*

```

select first_name, last_name
from (select first_name, last_name
from book_library.customers
order by first_name, last_name)
where rownum <6;

```

## 9 DDL and DML statements with transaction handling

**Example 113** *Create the house table with the following columns: id: numeric type, name: string, it has to be filled, date\_of\_build: date type, zip\_code fixed long string, price: numeric type. The primary key of the table is the id. The name should be unique in the table. The price should be more than 100 if it is filled.*

```

create table house
(h_id number(5),
name varchar2(30) not null,
date_of_build date,
zip_code char(4),
price number(8,2),
constraint h_pk primary key (h_id),
constraint h_uq unique (name),
constraint h_ch check (price>100)
);

```

**Example 114** *Put a few rows into the house table.*

```

insert into house (h_id, name, date_of_build, zip_code, price)
values (10,'Green_Label_House',to_date('2000.01.01_20:20','yyyy.mm.dd_hh24_mi'),'4028',120);

commit;

insert into house (h_id, name, date_of_build, zip_code, price)
values (10,'Green_Label_House',to_date('2000.01.01_20:20','yyyy.mm.dd_hh24_mi'),'4028',120);

insert into house (h_id, name, date_of_build, zip_code, price)
values (20,'Green_Label_House',to_date('2000.01.01_20:20','yyyy.mm.dd_hh24_mi'),'4028',120);

insert into house (h_id, name, date_of_build, zip_code, price)
values (20,'Red_Label_House',to_date('2000.01.01_20:20','yyyy.mm.dd_hh24_mi'),'4028',120);

commit;

insert into house (h_id, name, date_of_build, zip_code, price)

```

```
values (330,'Blue_Label_House',to_date('2000.01.01_20:20','yyyy.mm.dd_hh24_mi'),
'4028',10);
```

```
insert into house (h_id, name, date_of_build, zip_code, price)
values (330,'Blue_Label_House',to_date('2000.01.01_20:20','yyyy.mm.dd_hh24_mi'),
'4028',120.89);
commit;
```

**Example 115** *Delete the row from the house table which id is 330.*

```
delete from house
where h_id=330;
commit;
```

**Example 116** *Create the tenant table with the following columns: name: string house\_id: numeric, color\_of\_hair string. The primary key of the table is the house\_id and the name together. The house\_id is a foreign key which refers to the primary key of the house table.*

```
create table tenant
(name varchar2(30),
house_id number(5),
color_of_hair varchar2(30),
constraint te_pk primary key (house_id, name),
constraint te_fk foreign key (house_id) references house (h_id));
```

**Example 117** *Give new rows to the tenant table.*

```
insert into tenant (name, house_id, color_of_hair)
values ('John',10, 'red');
commit;
insert into tenant (name, house_id, color_of_hair)
values ('Jane',10, 'red');
commit;

insert into tenant (name, house_id, color_of_hair)
values ('Jane',10000, 'red');

insert into tenant (name, house_id, color_of_hair)
values ('Ann',10, null);
commit;
```

**Example 118** *List all information from the house table. If a house has any tenants list also this information.*

```
select *
from house h left outer join tenant t
on h.h_id=t.house_id;
```

**Example 119** *Change the color of hair to brown for the rows where the name is Jane.*

```
update tenant
set color_of_hair='brown'
where name='Jane';
commit;
```

**Example 120** *Create a table named tryout. It should have only one column: name which type is string.*

```
create table tryout
(name varchar2(30));
```

**Example 121** *Drop the tryout table.*

```
drop table tryout;
```

**Example 122** *Add a new column to the house table: house\_size which type is numeric.*

```
alter table house
add (house_size number(5));
```

**Example 123** *Drop the house\_size column of the house table.*

```
alter table house
drop column house_size;
```

**Example 124** *Drop the constraint called te\_fk of the tenant table.*

```
alter table tenant
drop constraint TE_FK;
```

**Example 125** *Add a new constraint to the tenant table: a foreign key which shows that the house\_id of the table refers to primary key of the house table.*

```
alter table tenant
add constraint TE_FK foreign key (house_id) references house(h_id);
```

**Example 126** *Drop the primary key of the tenant table.*

```
alter table tenant
drop primary key;
```

**Example 127** *Add a primary key to the tenant table: the house\_id and the name together.*

```
alter table tenant
add constraint te_pk primary key (house_id, name);
```

**Example 128** *Change the type of the name column of the tenant table: it should be maximum 50 character long string.*

```
alter table tenant
modify (name varchar2(50));
```

**Example 129** *Rename the name column of the tenant table to name2;*

```
alter table tenant
rename column name to name2;
```

**Example 130** *Rename the tenant table to tenant2;*

```
rename tenant to tenant2;
```

**Example 131** *Create a table which contains the title, isbn and price per page of the books which topic is thriller.*

```
create table my_books as
select price/number_of_pages price_per_pages, isbn,title
from book_library.books
where topic='Thriller';
```

**Example 132** *Create a view which lists the title, isbn and price per page of the books which topic is thriller.*

```
create view my_view as
select price/number_of_pages price_per_pages, isbn,title
from book_library.books
where topic='Thriller';
```

**Example 133** *Create a view which lists the last 10 borrowing with the customer name and the book\_title.*

```
create view task1 as
select borrowing_date, first_name, last_name, bks.TITLE
from (select *
from (select *
from book_library.borrowing
order by borrowing_date desc)
where rownum<11) bo
inner join book_library.customers cu
on bo.customer_id=cu.library_card_number
inner join book_library.book_items bit
on bo.book_item_id=bit.book_item_id
inner join book_library.books bks
on bit.book_id=bks.book_id;
```

## 10 DML statements

**Example 134** *Give a new row into the writing table in which Agatha Christie wrote the book with highest price. The royalty was 5000 ft.*

```
insert into my_writing (author_id, book_id, royalty)
select author_id, book_id, 5000
from book_library.authors, book_library.books
where last_name='Christie'
and first_name='Agatha'
and price=(select max(price) from book_library.books);
commit;
```

**Example 135** *At this moment the oldest student is borrowing the book-item which has the lowest value. Give a new row to the borrowing table.*

```
insert into my_borrowing(customer_id, book_item_id, borrowing_date)
select library_card_number, book_item_id, sysdate
from book_library.customers, book_library.book_items
where category='student'
and birth_date=(select min(birth_date)
from book_library.customers
```

```

        where category='student')
and THEORETICAL_VALUE=(select min(THEORETICAL_VALUE) from book_library.book_items);
commit;

```

**Example 136** *Change content of the table that way that the theoretical values of the book\_items should be multiplied by 10.*

```

update book_items
set theoretical_value=theoretical_value*10;
rollback;

```

**Example 137** *Change content of the table that way that the theoretical values of the book\_items should be increased by the price of the book where the book is published after 2000.*

```

update book_items bit
set theoretical_value=
theoretical_value+(select price
                    from book_library.books bo
                    where bo.book_id=bit.book_id)
where book_id in (select book_id
                  from book_library.books
                  where PUBLISHING_DATE>to_date('2000','yyyy'));

```

**Example 138** *Delete the rows from the writing table where the authors was born after 1900.*

```

delete
from writing
where author_id in (select author_id from book_library.authors
                    where birth_date >to_date('1900','yyyy'));
rollback;

```

**Example 139** *Delete the books which has no writing.*

```

delete
from books
where book_id not in (select book_id from book_library.writing);
commit;

```

**Example 140** *Insert a new row into the writing table in which Agatha Christie writes the book which has the highest price. She gets 2000 Ft royalty.*

```

insert into writing (author_id, book_id, royalty)
select author_id, book_id, 2000
from book_library.authors au, book_library.books
where au.last_name='Christie'
and first_name='Agatha'
and price=(select max(price) from book_library.books);
commit;

```

**Example 141** *Insert a new row into the borrowing table in which today the customer whose name is Williams Robert borrows the book\_item which has no theoretical value.*

```

insert into borrowing (customer_id,book_item_id,borrowing_date)
select library_card_number, book_item_id, sysdate
from book_library.customers, book_library.book_items bi
where last_name='Williams'
      and first_name='Robert' and bi.THEORETICAL_VALUE is null;
commit;

```

**Example 142** *Delete the authors who has no writing.*

```

delete --select *
from authors
where author_id not in (select author_id from book_library.writing);
commit;

```

**Example 143** *Delete the last borrowing.*

```

delete--select *
from borrowing
where borrowing_date=(select max(borrowing_date) from borrowing);

```

**Example 144** *Delete the book\_items which theoretical value is less than the price of the related book.*

```

delete--select *
from book_items bit
where theoretical_value<(select price
from book_library.books bo
where bit.book_id=bo.book_id);
commit;

```

**Example 145** *Update the theoretical value of the book items which topic is Thriller. The theoretical value should be increased by the price divided by the number\_of\_pages of the related book.*

```
update book_items bit
set theoretical_value=theoretical_value+(select price/number_of_pages
                                         from book_library.books bo
                                         where bit.book_id=bo.book_id)
where book_id in (select book_id from book_library.books where topic='Thriller');
commit;
```

**Example 146** *Decrease the royalty of the oldest author with the price of the connected book.*

```
update writing w
set royalty=royalty-(select price from book_library.books bo
                     where w.book_id=bo.book_id)
where author_id in (select author_id
                   from book_library.authors
                   where birth_date=(select min(birth_date) from book_library.authors));
commit;
```