APPENDIX 4. PROBLEMS FOR TESTS AND INTERVIEWS

A4.1 General DDL and DML

Create a database that will contain a table with fields: primary key (one field) with auto-increment; at least one text field; at least one numeric field; the number of hits, the default value is 0.

Fill in the created table with the initial data (5 rows).

Select all the records from the created table.

```
drop database if exists exam_task;
create database exam_task;

use exam_task;

create table product (
    id int primary key _____________________,
    title text not null,
    price _____________________ not null,
    num_accesses int not null _____________________ 0
);

_____________________ product (title, price)
_____________________ ("Phone", 100), ("iPod", 50), ("T-Shirt", 20),
("PC", 1000), ("Cola", 1);

select * _____________________ _____________________;
```

A4.2 Stored procedures and transactions

Create a stored procedure that has one argument – the number of table rows to print, while in the procedure body: initiate the beginning of a transaction that is completed or canceled manually; update the hit count field for all table entries, increasing the value by 1; if the specified number of rows is greater than the existing or equal to 0, the message “Invalid value” should be displayed and the transaction canceled; otherwise, you should display the specified number of rows (from the beginning), confirm the transaction, and display the message “Returned X rows”, where X is the specified number. Call procedure with parameters -3, 0, 4, 8.

```
delimiter //
create _____ print_products
(_____ records int)
begin
    declare products _____;

    set autocommit = _____;
    start _____;

    update _____ set num_accesses =
_____;
    select count(*) into _____ from
_____;

    if _____ _____
_____ then
        begin
            select "Invalid values";
            _____;
```

```
            end;
        else
            begin
                select * from _____________________
_____________________ records;
                select concat(records, " rows returned.");

                _____________________;
            end;
        end if;
    end //

    _____________________ ;

    call print_products(-3);
    call print_products(0);
    call print_products(4);
    call print_products(8);

    drop _____________________ exam_task;
```