

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КРЫМ**

**«КРЫМСКИЙ ИНЖЕНЕРНО-ПЕДАГОГИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ ФЕВЗИ ЯКУБОВА»**

**Кафедра прикладной информатики**

**Методические указания по выполнению лабораторных работ  
по дисциплине  
«БАЗЫ ДАННЫХ»**

Направление подготовки  
**09.03.03 Прикладная информатика**

Профиль  
Прикладная информатика в информационной сфере

Факультет экономики, менеджмента и информационных технологий

Симферополь, 2021

**Тема «Анализ требований для ИС и выделение сущностей предметной области»**

Кол.-во часов: 2

**Ход работы**

*Часть 1.* Предположим, что необходимо создать информационную систему, которая хранит данные о книгах, данные об издателе, данные издания и тд. Ниже представлен макет БД «Bookbiz»

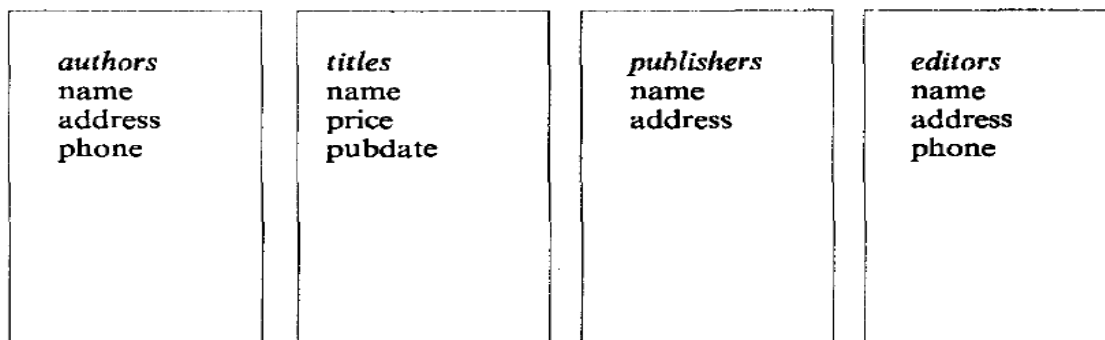


Рис. 1. макетБД «Bookbiz»

**Добавитьтаблицу “Titleauthors”**

*Часть 2.* Индивидуальное задание – выбрать тему по варианту, описать требования к системе, выделить и описать основные сущности предметной области.

**Варианты**

1. Книжный электронный магазин.
2. Продажа флеш-накопителей через Интернет.
3. Продажа продуктов питания через Интернет.
4. Продажа видеофильмов через Интернет.
5. Спортивный электронный магазин.
6. Цветочный электронный магазин.
7. Продажа и бронирование ж/д билетов через Интернет.
8. Продажа и бронирование авиабилетов через Интернет.
10. Прокат видеофильмов через Интернет
11. Прокат спортивного инвентаря через Интернет
12. Интернет - аукцион.
13. Букмекерский Интернет - сервис.
15. Туристический сервис через Интернет.
16. Гостиничный сервис через Интернет.
17. Сайт объявлений и деловых предложений.
18. Реализация представительства компании в Интернет (с возможностью просмотра каталога продукции и её заказа через Интернет).
19. Информационная система по сдаче автомашин внаем.
20. Информационные системы, ускоряющие потоки товаров.
21. Информационная система банка

**Литература:**

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

**Инструменты:**


1. MySQLWorkbench
2. MySQL

## Тема «Создание EER -диаграммы»

Кол.-во часов: 4

## Ход работы

**1. Разработка проекта и баз данных**

Создание проекта приложения происходит в среде MySQLWorkbench. На домашнем экране приложения необходимо кликнуть на иконку  рядом с секцией Models в нижней части страницы (рис. 2.1), либо выбрать пункт New Model меню File (рис. 2.2).

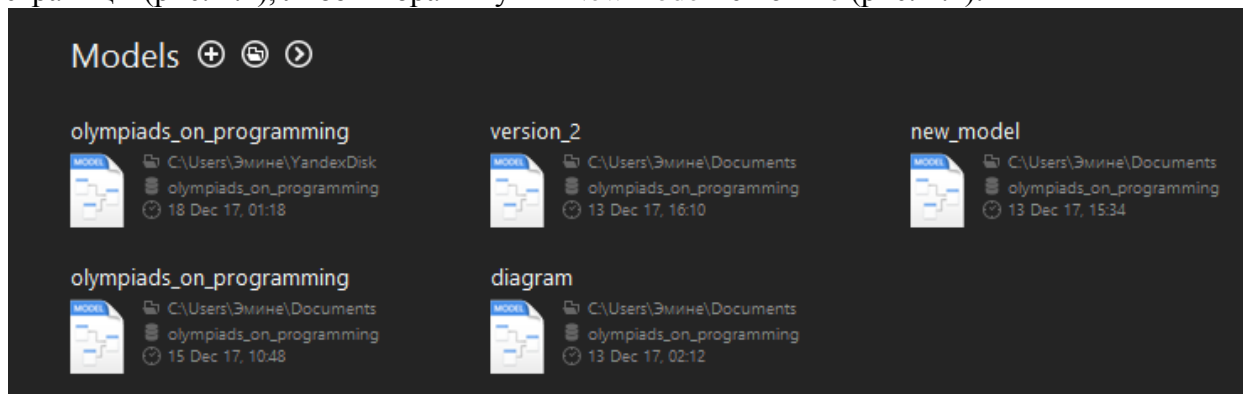


Рис. 2. Секция Models домашнего экрана приложения MySQLWorkbench

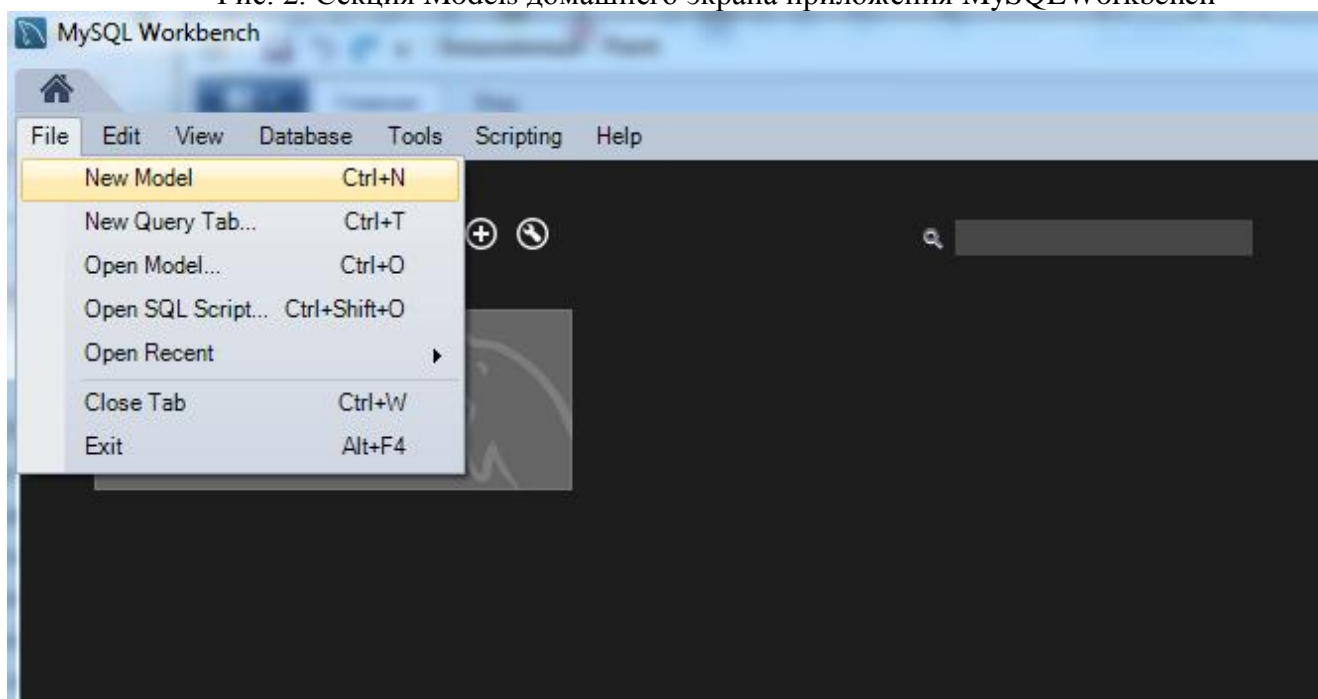



Рис. 3. Домашний экран приложения MySQLWorkbench

Для добавления схемы необходимо нажать на кнопку , расположенную в правой части панели инструментов PhysicalSchemas. Имя по умолчанию для схемы – «new\_schema1», оно было изменено на «Bookbiz» путем модификации поля Name схемы.

**2. Создание таблиц, заполнение их тестовой информацией**

Была создана пустая Enhanced Entity-Relationship (EER) диаграмма. Для создания пустой EER диаграммы в MySQLWorkbench необходимо кликнуть на иконку AddDiagram окна ModelOverview. При помощи панели инструментов, расположенной слева в окне Diagram, была добавлена новая таблица.

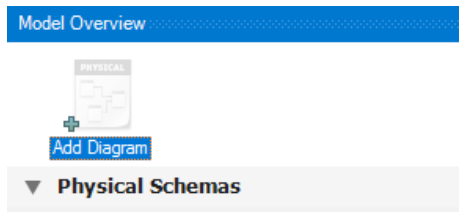


Рис. 4. Окно добавления диаграммы

После добавления таблицы в EER-диаграмму происходит загрузка редактора таблиц с именем таблицы по умолчанию table1. Поле TableName было отредактировано, изменено название таблицы с «table1» на «publishers». В созданную таблицу необходимо добавить поля. Двойной клик по ячейке колонки ColumnName позволяет добавить новое поле в таблицу.

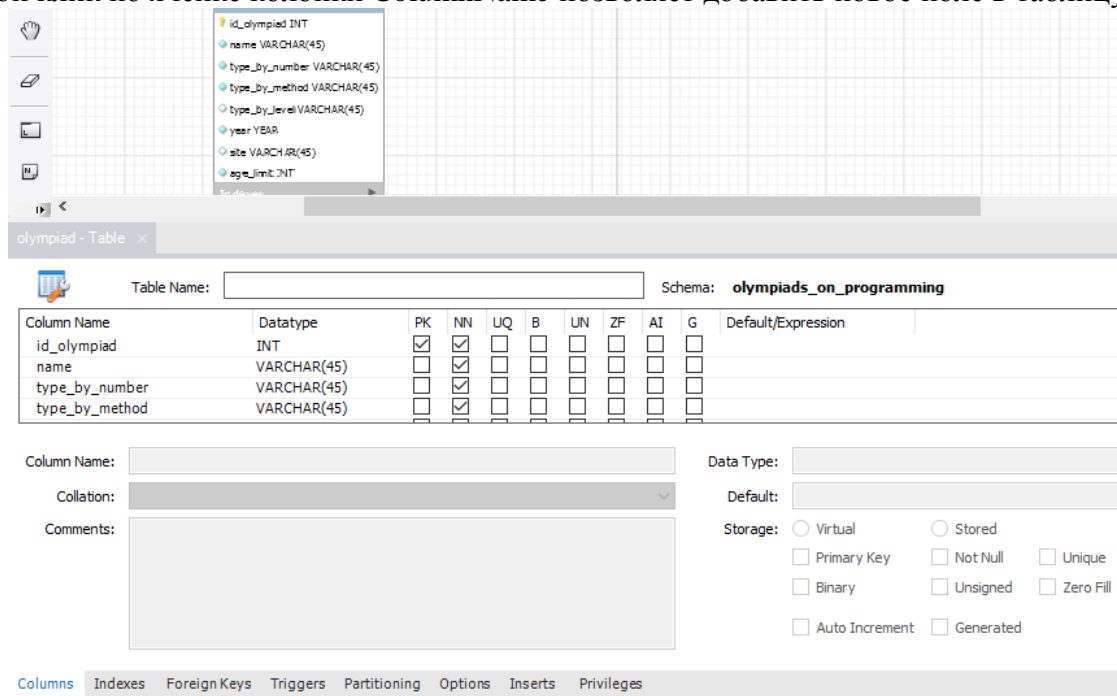


Рис. 5. Окно свойств таблицы

Аналогично создаются и все остальные таблицы. Далее необходимо создать связи между таблицами. Связи между таблицами создавались с помощью внешних ключей. После указания необходимых внешних ключей связи создаются автоматически. Также работает и обратная связь, то есть можно воспользоваться панелью инструментов, а создание внешних ключей будет произведено автоматически. Для редактирования связи необходимо воспользоваться пунктом EditRelationship контекстного меню связи.

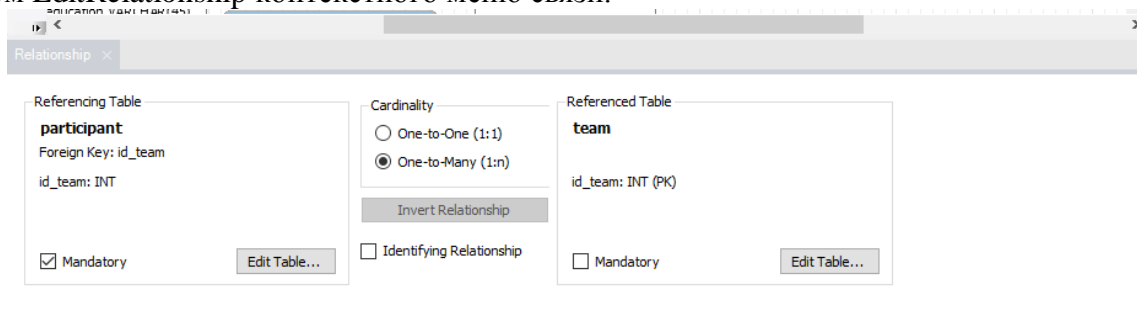


Рис. 6. Окно редактирования связей

Данный пункт позволяет изменять название, тип связи, а также класс принадлежности. Если отношение имеет необязательный класс принадлежности, необходимо снять галочку с флажка «Mandatory» в группе элементов ReferencingTable.



Для наглядности представлена студенческая EER диаграмма для предметной области «Олимпиады по программированию» на рисунке 7.

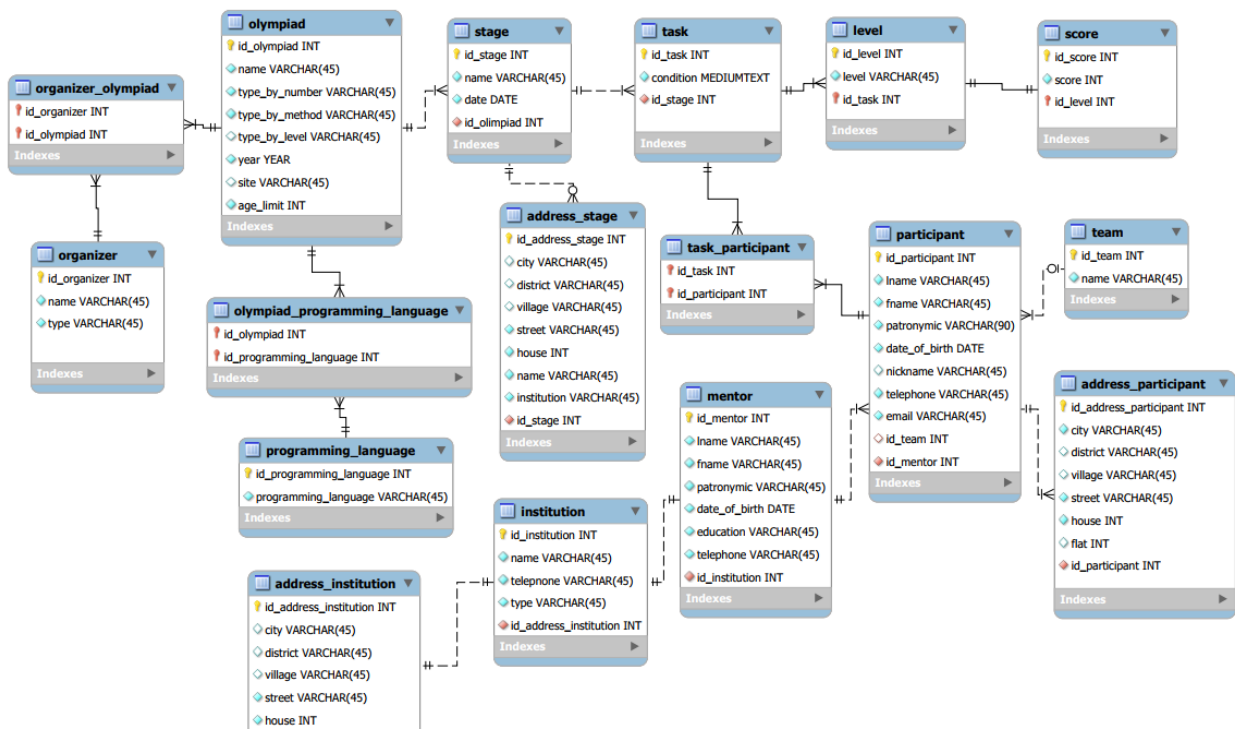


Рис. 7. EER диаграмма базы данных «Олимпиады по программированию»

Индивидуальное задание –создать для своей темы по вариантам EER модель, используяMySQLWorkbench

#### **Литература:**

- Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

#### **Инструменты:**

- MySQLWorkbench
- MySQL

## Тема «Синхронизация EER диаграммы в среду MySQL»

Кол.-во часов: 4

## Ход работы

После создания модели синхронизировали ее с базой данных MySQL. Для этого необходимо воспользоваться пунктом Synchronize Model меню Database (рис.8.). Данный пункт запускает мастер Synchronize Model with Database. Окно Connection Options позволяет выбрать подключение, а также задать дополнительные опции для выбранного соединения. Поскольку в данном случае нет необходимости изменять настройки соединения, выполнили переход к следующей странице, нажав на кнопку Next.

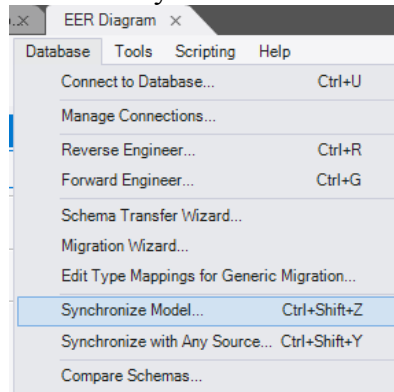


Рис.8. Меню Database

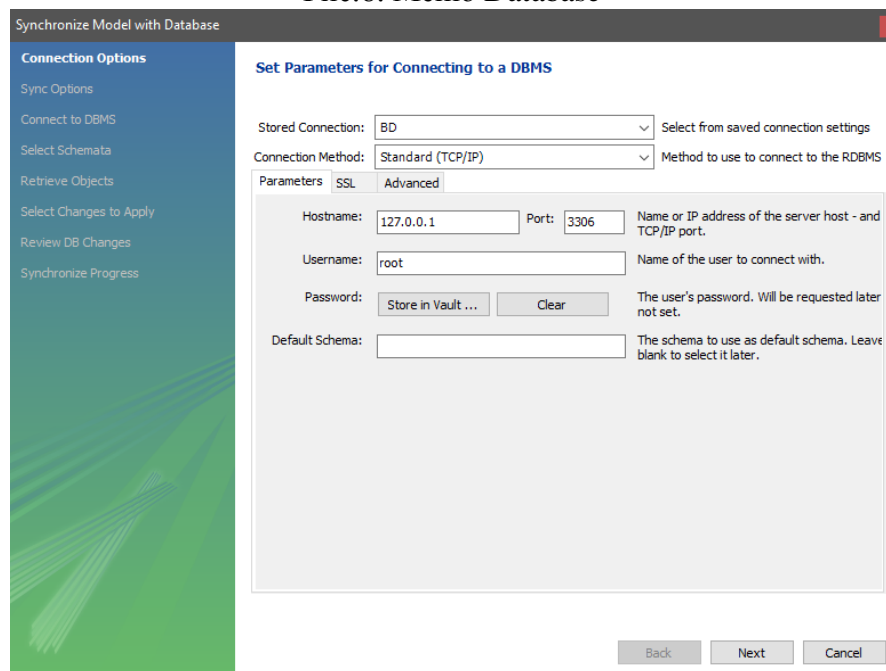


Рис. 9. Страница Connection Options мастера Synchronize model with Database

Страница Options мастера содержит опциональные расширенные настройки. Настройки по умолчанию являются приемлемыми для данной базы данных, поэтому осуществляется переход к следующей странице мастера, в которой нужно выбрать схему БД с которой будет синхронизирована модель (рис.10). В данном случае происходит синхронизация количества всех объектов, которыми являются таблицы (рис.11).

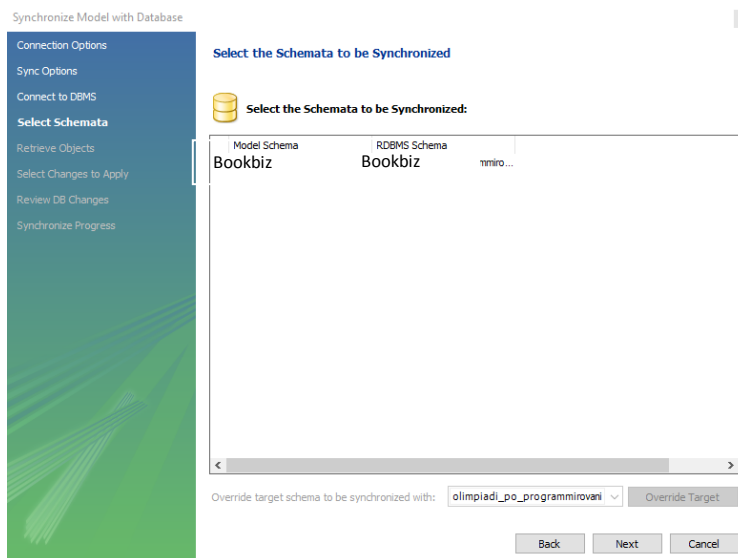


Рис. 10. Страница Select Schemata мастера Synchronize model with Database

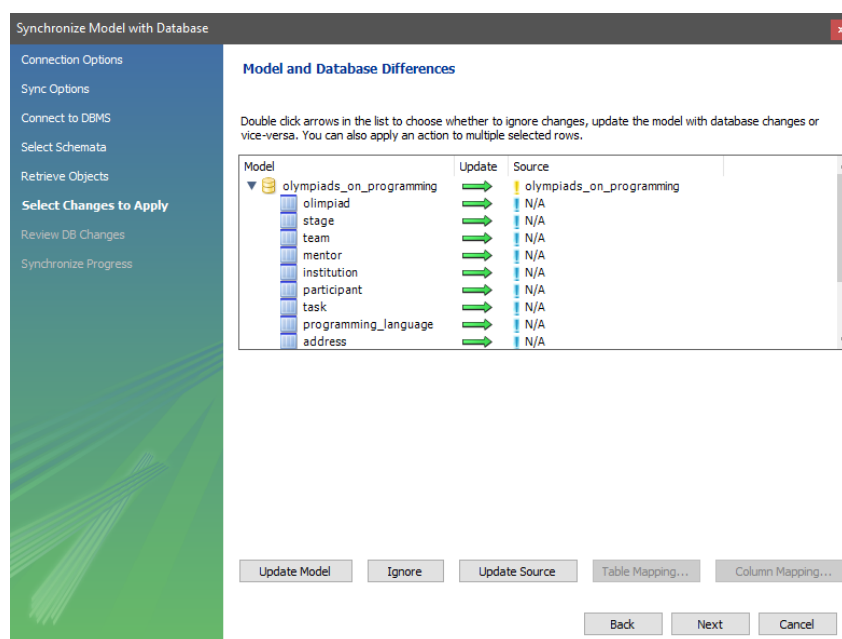


Рис. 11. Страница Select Changes to Apply мастера Synchronize model with Database

Окно Review SQL Script отображает SQL-сценарий, который будет выполнен на стороне сервера для создания схемы «bookbiz». Сценарий содержит запросы на создание таблиц. Кнопки Save to File и Save to Clipboard позволяют сохранить сценарий в файл и буфер обмена, соответственно. На данном этапе в SQL-сценарий возможно вносить необходимые поправки.



Ниже приведен пример студенческой работы по созданию таблицы на стороне сервера для БД «Олимпиады по программированию»

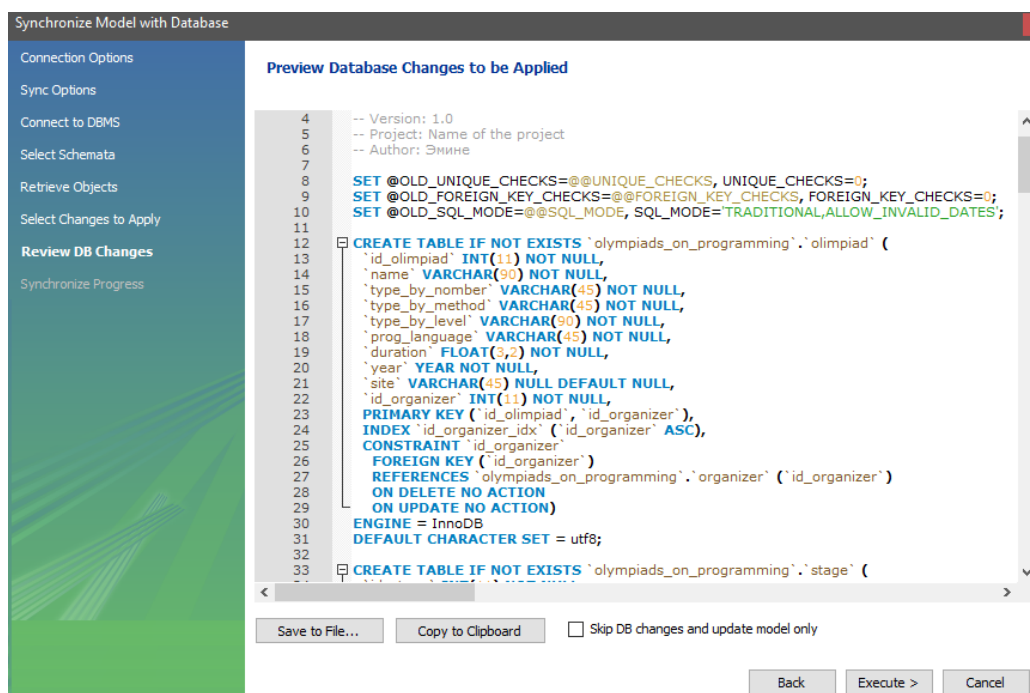


Рис. 12. SQL сценарий, позволяющий создать таблицы на стороне сервера



В результате получили схему БД по EER диаграмме (Рис.13.)

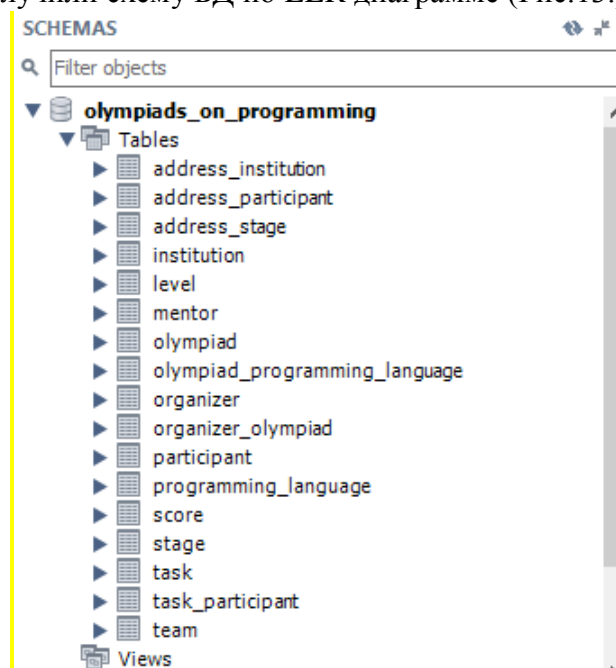


Рис.13. СхемаБД «olympiads\_on\_programming»

Индивидуальное задание – синхронизировать EER диаграмму, полученную в ходе реализации темы по вариантам, в среду MySQL.

### **Литература:**

3. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

### **Инструменты:**

1. MySQLWorkbench
2. MySQL



**Тема «Тема «Создание запросов на языке SQL.  
Операторы CREATE, INSERT, UPDATE, DELETE»**

Кол.-во часов: 4

**Ход работы**

Создайте БД «BookBiz», в ней с использованием команды CREATE создайте таблицу Publisher и заполните данными, как на рисунке (поле pub\_id сделайте *ключевым*)

pub_id	pub_name	address	city	state
0736	New Age Books	1 1st St	Boston	MA
0877	Binnet & Hardley	2 2nd Ave.	Washington	DC
1389	Algodata Infosystems	3 3rd Dr.	Berkeley	CA

Выведите на экран содержимое таблицы, используя запрос  
SELECT \* FROM Publishers

Добавить в таблицу еще одну запись, используя INSERT (при необходимости укажите перечень названий полей, в которые вы хотите сделать вставку)

```
insert into publishers
values ('0010', 'Pragmatics', '4 4th Ln.', 'Chicago', 'IL')
```

Выведите на экран то, что получилось. Должно получиться так

pub_id	pub_name	address	city	state
0736	New Age Books	1 1st St	Boston	MA
0877	Binnet & Hardley	2 2nd Ave.	Washington	DC
1389	Algodata Infosystems	3 3rd Dr.	Berkeley	CA
0010	Pragmatics	4 4th Ln.	Chicago	IL

**Задание.** а) Используя команду UPDATE измените записи в таблице Publisher согласно рисунку

```
update publishers
set city = 'Atlanta', state = 'GA'
```

Посмотрите, что получилось

Теперь попробуйте обновить только одну запись, подумайте, как это сделать.

Добавьте еще пять записей в таблицу

б) Используя команду DELETE, удалите последнюю запись в таблице Publisher.

```
delete from publishers
where pub_id = '1622'
```

Результаты запросов выведите на экран



С помощью языка запросов SQL были добавлены данные в таблицы. Примеры вставки данных представлены на рисунке 2.13.

```

INSERT INTO `olympiads_on_programming`.`organizer` (`id_organizer`, `name`, `type`)
VALUES('4', 'Яндекс', 'IT-компания');

INSERT INTO `olympiads_on_programming`.`olympiad` (`id_olympiad`, `name`, `type_by_number`, `type_by_method`, `type_by_level`, `year`, `site`, `age_limit`)
VALUES('1', 'Яндекс.Алгоритм', 'Командная', 'Онлайн', NULL, 2016, 'https://contest.yandex.ru/algorithm2016/', '18');

INSERT INTO `olympiads_on_programming`.`programming_language` (`id_programming_language`, `programming_language`)
VALUES('1', 'Pascal');

INSERT INTO `olympiads_on_programming`.`stage` (`id_stage`, `name`, `date`, `id_olympiad`)
VALUES('1', 'Отборочный', '2016-06-04', '1');

INSERT INTO `olympiads_on_programming`.`participant` (`id_participant`, `lname`, `fname`, `patronymic`, `date_of_birth`, `nickname`, `telephone`, `email`, `id_team`, `id_mentor`)
VALUES('1', 'Крылов', 'Петр', 'Сергеевич', '1999-08-23', NULL, '79787805469', 'krilov.99@gmail.com', NULL, '1');

INSERT INTO `olympiads_on_programming`.`organizer` (`id_address_participant`, `city`, `district`, `village`, `street`, `house`, `flat`, `id_participant`)
VALUES('1', 'Евпатория', NULL, NULL, 'Новиков', '12', '9', '0');

INSERT INTO `olympiads_on_programming`.`task` (`id_task`, `condition`, `id_stage`)
VALUES('1', 'Поскольку множества гласных и согласных не пересекаются, при лексикографическом сравнении раньше будет та последовательность, которая начинается с буквы, которая р

```

Рис.14 Добавление данных в таблицы.

Индивидуальное задание – создать и заполнить данными таблицы для своего варианта

### **Литература:**

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

### **Инструменты:**

1. MySQLWorkbench
2. MySQL

## Лабораторная работа 5 Тема «Создание запросов на языке SQL. Операция WHERE»

Кол.часов – 2

### Ход работы Упрощенный синтаксис

*SELECT* список полей выбора  
*FROM* список таблиц  
*WHERE* условие поиска

На основе таблицы Publisher создайте запрос типа

*SELECT* \*  
*FROM* Publisher  
*WHERE* state='CA'

### **Результат:**

pub_id	pub_name	address	city	state
1389	Algodata Infosystems	3 3rd Dr.	Berkeley	CA

Создайте еще одну таблицу Titles полями – **title id** (ключевое), title, type, pub\_id, price, advance, ytd\_sales, contract, notes, pubdate

```

create table titles
(title_id char(6) not null,
title varchar(80) not null,
type char(12) null,
pub_id char(4) null,
price money null,
advance money null,
ytd_sales int null,
contract bit not null,
notes varchar(200) null,
pubdate datetime null)

```

Заполните созданную таблицу данными. И реализуйте следующий запрос

```
select title, pub_name
from titles, publishers
where publishers.pub_id = titles.pub_id
```

Далее реализуйте запрос для таблицы Titles

```
update titles
set price = price * 2
```

Индивидуальное задание –разработать запросы для своего варианта, используя оператор WHERE на языке SQL.

**Литература:**

2. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

**Инструменты:**

3. MySQLWorkbench
4. MySQL

**Лабораторная работа 6**

**Тема «Создание запросов на языке SQL. Выборка»**

Кол.часов – 2

**Ход работы**

1. Выполните запрос, как на рисунке

```
SELECT title_id, publishers.*
from titles, publishers
where titles.pub_id = publishers.pub_id
```

Результат:

title_id	pub_id	pub_name	address	city	state
PS1372	0736	New Age Books	1 1st St	Boston	MA
PS7777	0736	New Age Books	1 1st St	Boston	MA
PS3333	0736	New Age Books	1 1st St	Boston	MA
PS2106	0736	New Age Books	1 1st St	Boston	MA
PS2091	0736	New Age Books	1 1st St	Boston	MA
BU2075	0736	New Age Books	1 1st St	Boston	MA
MC3026	0877	Binnet & Hardley	2 2nd Ave.	Washington	DC
MC2222	0877	Binnet & Hardley	2 2nd Ave.	Washington	DC
TC7777	0877	Binnet & Hardley	2 2nd Ave.	Washington	DC

2. Далее выполните запрос на выборку для таблицы Publisher

SQL:

```
select pub_id, pub_name
from publishers
```

Результат:

pub_id	pub_name
0736	New Age Books
0877	Binnet & Hardley
1389	Algodata Infosystems

SQL:

```
select pub_name, pub_id
from publishers
```

Результат:

pub_name	pub_id
New Age Books	0736
Binnet & Hardley	0877
Algodata Infosystems	1389

3. Далее выполните запрос на выборку для таблицы Titles

SQL:

```
select title, advance*2 as Double_Advance
from titles
```

Выведите результат на экран

4. Выполните запрос на умножение двух столбцов, как на рисунке

SQL:

```
select title_id, ytd_sales * price
from titles
```

Результат:

title_id	
PC8888	81900.0000
BU1032	81859.0500
P57777	26654.6400
P53333	81399.2800
BU1111	46318.2000
MC2222	40619.6800
TC7777	61384.0500
TC4202	180307.2000

5. Еще один запрос с использованием математических операций

```
select title_id, ytd_sales * (price - advance)
from titles
```

Результат:

title_id	
PC8888	-32678100.000
BU1032	-20393140.950
P57777	-13317345.360
P53333	-8062600.720
BU1111	-19333681.800
MC2222	40619.680
TC7777	-32698615.950
TC4203	-60203602.800
PC1035	-61258499.000
MC3026	(NULL)

Индивидуальное задание – разработать запросы для своего варианта по выборке на языке SQL.

#### **Литература:**

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

#### **Инструменты:**

1. MySQLWorkbench
2. MySQL

### **Лабораторная работа 7**

#### **Тема «Создание запросов на языке SQL. Логические операции»**

Кол. часов – 2

#### **Ход работы**

В языке имеется ряд операторов и ключевых слов для задания логических условий.

1. (=, >, <, ...)  
WHERE advance\*2>ytd\_sales\*price
2. (AND, NOT, OR)  
WHERE advance<5000 or ytd\_sals>2000
3. (BETWEEN, NOT BETWEEN)  
WHERE ytd\_sales BETWEEN 4095 AND 1200
4. (IN, NOT IN) – списки  
WHERE state IN ('CA', 'IN', 'MD')
5. (IS NULL, IS NOT NULL) – неизвестные значения  
WHERE advance IS NULL
6. (LIKE, NOT LIKE) – соответствия символов  
WHERE pub\_id NOT LIKE '0010'

Символ	Значение
LIKE '27%'	27, за которым следует любая строка символов
LIKE '27@%'	27%
LIKE '_n'	an, in, on и так далее
LIKE '@_n'	_n

**Задание.** Проверьте работу логических операторов на примерах таблицы Titles, ,6, выполните ряд запросов, как на рисунках

1)

```
select title, type, price, advance
from titles
where type = 'business'
and price > $10.00
and advance < $20000
```

2)

```
select title, type, price advance
from titles
where type = 'business'
or price > $10.00
or advance < $20000
```

3)

```
select title_id, type, advance
from titles
where (type = 'business'
or type = 'psychology')
and advance > $5500
```

4)

```
select title_id, type, price, advance, ytd_sales
from titles
where price * ytd_sales < 2 * advance
and pubdate < '10/15/85'
```

5)

```
select title_id, notes
from titles
where notes like '%exercise%'
```

**Задание.** Разработайте однотабличные и много табличные запросы, используя операторы (BETWEEN, NOTBETWEEN), (IN, NOTIN) – списки, (ISNULL, ISNOTNULL) – неизвестные значения, применив к таблицам Editors, Tittles, Publishers.

Индивидуальное задание – разработать запросы для своего варианта, используя логические операции на языке SQL.

#### **Литература:**

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

#### **Инструменты:**

1. MySQLWorkbench
2. MySQL

## **Лабораторная работа 8** **Тема «Агрегирующие функции. Группировка»**

Кол. часов – 2

### **Ход работы**

В стандарте языка SQL определено несколько агрегатных функций:

- COUNT — возвращает количество значений в указанном столбце;
- SUM — возвращает сумму значений в указанном столбце;
- AVG — возвращает усредненное значение в указанном столбце;
- MIN — возвращает минимальное значение в указанном столбце;
- MAX — возвращает максимальное значение в указанном столбце.

Выполните следующие запросы

1)

SQL:

```
select sum(ytd_sales) as Total
from titles
```

2)

SQL:

```
select avg(price * 2)
from titles
```

3)

SQL:

```
select count(price), count(*)
from titles
```

4)

SQL:

```
select min(au_lname)
from authors
```

5)

SQL:

```
select count(price), sum(distinct price)
from titles
```

### **Группировка**

1)

SQL:

```
select pub_id, type, count(type)
from titles
group by pub_id, type
```

2) расширение

SQL (расширение):

```
select pub_id, type, title_id
from titles
order by pub_id, type
compute count (title_id) by pub_id, type
compute count (title_id) by pub_id
```

3)

SQL:

```
select type, avg(advance), sum(ytd_sales)
from titles
group by type
```

4)

SQL:

```
select type, avg(price)
from titles
where advance > $5000
group by type
```

Индивидуальное задание – разработать запросы для своего варианта, используя агрегирующие функции на языке SQL.

### **Литература:**

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

### **Инструменты:**

1. MySQLWorkbench
2. MySQL

**Лабораторная работа №9**  
**Тема «Предикат HAVING, функция ISNULL()»**

Кол. часов – 2

Ход работы

Выполните следующие запросы

1)

```
SQL:
select type, count(*)
from titles
group by type
having count(*) > 1
```

2)

```
SQL:
select pub_id, sum(advance), avg(price)
from titles
group by pub_id
having sum(advance) > $15000
and avg(price) < $20
and pub_id > '0800'
```

3)

```
SQL:
select pub_id, sum(advance), avg(price)
from titles
where price >= $5
group by pub_id
having sum(advance) > $15000
and avg(price) < $20
and pub_id > '0800'
order by pub_id
```

4)

```
SQL (вариант)
select isnull(type, 'What?'), count(*)
from titles
group by type
```

5)

```
SQL (вариант):
select title_id, isnull(advance, 4000), isnull(price, 35.00)
from titles
where advance < $6000 or advance is null
order by price
```

Индивидуальное задание – разработать запросы для своего варианта, используя предикат HAVING, функция ISNULL() на языке SQL.

**Литература:**

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

**Инструменты:**

1. MySQLWorkbench
2. MySQL



**Лабораторная работа №10**  
**Тема «Объединения»**

Кол.часов – 2

Ход работы

Выполните следующие запросы

1)

SQL:

```
select ed_lname, ed_id, editors.city, pub_name, publishers.city
from editors, publishers
where editors.city = publishers.city
and pub_name = 'Algodata Infosystems'
```

2)

SQL:

```
select ed_lname
from editors, publishers
where editors.city = publishers.city
and pub_name = 'Algodata Infosystems'
```

3)

SQL:

```
select au_lname, au_fname
from authors a, publishers p
where a.city = p.city
```

4)

SQL:

```
select au_lname, au_fname, title
from authors a, titles t, titleauthors ta
where a.au_id = ta.au_id
and t.title_id = ta.title_id
and t.type = 'trad_cook'
```

5)

SQL:

```
select au_fname, au_lname, pub_name
from authors, publishers
where authors.city = publishers.city
```

6) Оператор объединения UNION

SQL:

```
select au_fname, au_lname, city
from authors
where city in ('Oakland', 'Berkeley')
union
select ed_fname, ed_lname, city
from editors
where city in ('Oakland', 'Berkeley')
```

Индивидуальное задание – разработать запросы для своего варианта, используя предикат HAVING, функция ISNULL() на языке SQL.

**Литература:**

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

#### Инструменты:

1. MySQLWorkbench
2. MySQL

### Лабораторная работа №11

#### Тема «Структурированные запросы и подзапросы»

Кол. часов – 4

Ход работы

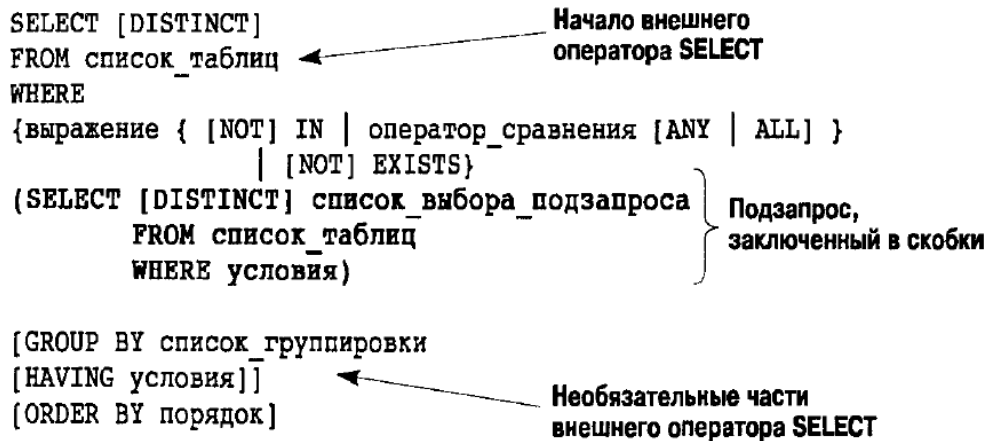


Рис. 15. Упрощенный синтаксис подзапроса

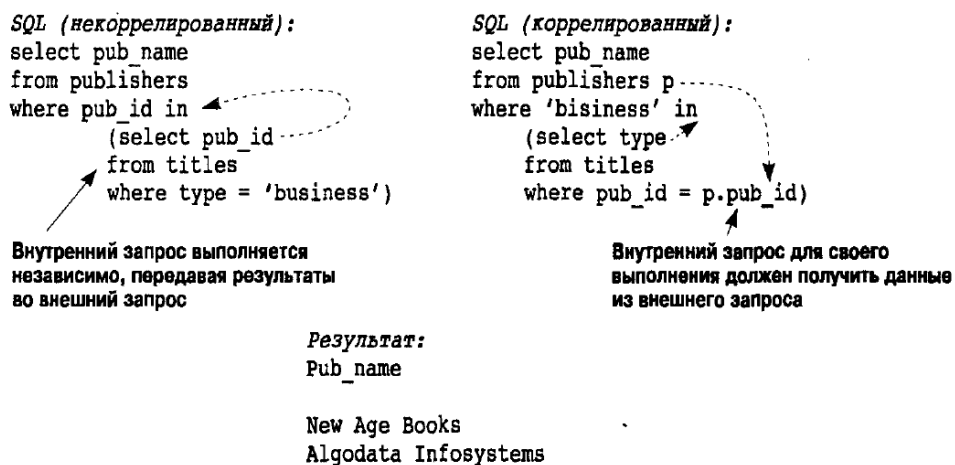


Рис. 16. Упрощенный синтаксис подзапроса - 2

Выполните следующие запросы

1)

SQL:

```
select pub_name
from publishers
where publishers.pub_id in
(select titles.pub_id
from titles
where type = 'business')
```

2)

SQL (коррелируемый):

```
select pub_name
from publishers
where 'business' in
(select type
 from titles
 where titles.pub_id = publishers.pub_id)
```

3)

SQL:

```
select title, price
from titles
where price =
      (select min(price)
       from titles)
```

4)

SQL:

```
select au_lname, au_fname, city
from authors
where city in
(select city
 from authors
 where au_fname = 'Livia'
 and au_lname = 'Karsen')
```

5) *подзапросы с IN*

SQL:

```
select au_lname, au_fname
from authors
where state = 'CA'
and au_id in
      (select au_id
       from titleauthors
       where royaltyshare < .30
       and au_ord = 2)
```

6)

SQL:

```
select authors.au_id, au_lname, au_fname
from authors, titleauthors
where royaltyshare < 1.0
and authors.au_id = titleauthors.au_id
and authors.au_id in
      (select distinct authors.au_id
       from authors, titleauthors
       where titleauthors.royaltyshare = 1.0
       and authors.au_id = titleauthors.au_id)
```

7)

SQL:

```
select distinct au_lname, au_fname
from authors
where 1.00 in
(select royaltyshare
 from titleauthors
 where au_id = authors.au_id)
```

8) ALL, ANY

ALL	Результат	ANY	Результат
>ALL(1, 2, 3)	>3	>ANY(1, 2, 3)	>1
< ALL(1, 2, 3)	<1	< ANY(1, 2, 3)	<3
= ALL(1, 2, 3)	=1 или =2 или =3	=ANY(1, 2, 3)	=1 или =2 или =3

SQL:

```
select title
from titles
where advance > all
    (select advance
     from publishers, titles
     where titles.pub_id = publishers.pub_id
     and pub_name = 'Algodata Infosystems')
```

9)

SQL:

```
select au_lname, au_fname
from authors
where city <> any
    (select city
     from publishers)
```

Индивидуальное задание – разработать подзапросы для своего варианта языка SQL.

**Литература:**

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

**Инструменты:**

1. MySQLWorkbench
2. MySQL

**Лабораторная работа №12**

**Тема «Подзапросы, возвращающие единственное значение»**

Кол. часов – 2

Ход работы

Выполните следующие запросы

1)

```
SQL:
select title
from titles
where price >
      (select min(price)
       from titles)
```

2)

```
SQL:
select title, type
from titles
where price >
      (select min(price)
       from titles
       group by type
       having type = 'trad_cook')
```

3) подзапросы, выполняющие проверку на существование

Начало операторов SELECT, INSERT, UPDATE, DELETE или подзапроса  
WHERE [NOT] EXISTS (подзапрос)  
[Конец операторов SELECT, INSERT, UPDATE, DELETE или подзапроса]

```
SQL:
select distinct pub_name
from publishers
where exists
      (select *
       from titles
       where pub_id = publishers.pub_id
       and type = 'business')
```

4)

```
SQL:
select pub_name
from publishers
where not exists
      (select *
       from titles
       where pub_id = publishers.pub_id
       and type = 'business')
```

5)

```
SQL:
select distinct city
from authors
where exists
      (select *
       from publishers
       where authors.city = publishers.city)
```

Индивидуальное задание – разработать соответствующие подзапросы для своего варианта на языке SQL.

#### ***Литература:***

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

**Инструменты:**

1. MySQLWorkbench
2. MySQL

**Лабораторная работа №13****Тема «Подзапросы с разными уровнями вложения»**

Кол. часов – 2

Ход работы

Выполните следующие запросы

1)

SQL:

```
select au_lname, au_fname
from authors
where au_id in
      (select au_id
       from titleauthors
       where title_id in
        (select title_id
         from titles
         where type = 'popular_comp') )
```

2)

SQL:

```
update titles
set price = price * 2
where pub_id in
      (select pub_id
       from publishers
       where pub_name = 'New Age Books')
```

3)

SQL:

```
delete salesdetails
where title_id in
      (select title_id
       from titles
       where type = 'business')
```

Индивидуальное задание – разработать соответствующие подзапросы для своего варианта на языке SQL.

**Литература:**

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

**Инструменты:**

1. MySQLWorkbench
2. MySQL

**Лабораторная работа №14****Тема «Поиск данных заданного размера»**

Кол. часов – 2

Ход работы

Выполните следующие запросы

1)

SQL:

```
select au_fname
from authors
where au_fname like '[A-Z] [a-z] [a-z] [a-z]'
```

2)

SQL:

```
select au_fname, phone, address, city, state, zip
from authors
where au_id like '123%'
```

3)

SQL:

```
select authors.city acity, authors.state astate, publishers.city pcity,
publishers.state pstate
from authors, publishers
where authors.city != publishers.city
      and authors.state != publishers.state
```

4)

SQL:

```
select authors.city, authors.state
from authors
where exists
(select *
 from publishers
 where publishers.city = authors.city
       and publishers.state = authors.state)
```

5)

SQL:

```
select title, title_id, price
from titles
where price between $7.00 and $10.95
```

6)

SQL:

```
select title, title_id, price
from titles
where price between
      (select price
       from titles
       where title like 'Life Without%')
and
      (select price
       from titles
       where title like 'Is Anger%')
```

7)

SQL:

```
select title, title_id, price
from titles
where price between
        (select min(price)
         from titles
         where title_id in ('PS2106', 'PS2091'))
and
        (select max(price)
         from titles
         where title_id in ('PS2106', 'PS2091'))
```

Индивидуальное задание – разработать соответствующие подзапросы для своего варианта на языке SQL.

***Литература:***

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

***Инструменты:***

1. MySQLWorkbench
2. MySQL

**Лабораторная работа №15**  
**Тема «Создание курсоров»**

Кол. часов – 4

Ход работы

Вот упрощенный синтаксис оператора, определяющего курсор:

```
CREATE VIEW имя_курсора [(имя_столбца [, имя_столбца]...)]
AS
SELECT_оператор
```

В следующем примере создается курсор, который отображает фамилии авторов, живущих в Окленде и Калифорнии, а также названия их книг:

SQL:

```
create view oaklanders
as
select au_fname, au_lname, title
from authors, titles, titleauthors
where authors.au_id = titleauthors.au_id
and titles.title_id = titleauthors.title_id
and city = 'Oakland'
```

Вот результат запроса:



SQL:

```
select *  
from oaklanders
```

Результат:

au_fname	au_lname	title
Marjorie	Green	The Busy Executive's Database Guide
Marjorie	Green	You Can Combat Computer Stress!
Dick	Straight	Straight Talk About Computers
Livia	Karsen	Computer Phobic and Non-Phobic Individuals: Behavior Variations
Stearns	MacFeather	Computer Phobic and Non-Phobic Individuals: Behavior Variations
Stearns	MacFeather	Cooking with Computers: Surreptitious Balance Sheets

---

## ПРЕИМУЩЕСТВА КУРСОРОВ

Чтобы показать преимущества использования курсоров, предположим, что у базы данных *bookbiz* есть несколько пользователей с разными профессиональными интересами. Допустим, что руководитель отдела рекламы хочет знать, какие авторы с какими книгами связаны и кто из них указан на обложке первым, кто вторым и кто третьим. Цены, объемы продаж, затраты, авторские гонорары и личные адреса авторов руководителя отдела рекламы не интересуют, но ему нужна определенная информация из каждой следующих трех таблиц: *titles*, *authors* и *titleauthors*. Если не пользоваться курсором, то для этого можно было бы применить запрос примерно следующего вида:

---

Создадим курсор

SQL:

```
create view books  
as  
select titles.title_id, au_ord, au_lname, au_fname  
from authors, titles, titleauthors  
where authors.au_id = titleauthors.au_id and  
titles.title_id = titleauthors.title_id
```

И выполним для него запрос

Теперь руководитель отдела рекламы может воспользоваться этим курсором для получения тех же результатов, не задумываясь об объединениях, списках выбора и условиях поиска:

SQL:

```
select *
from books
```

Результат:

title_id	au_ord	au_lname	au_fname
BU1032	1	Bennet	Abraham
PS7777	1	Locksley	Chastity
PC9999	1	Locksley	Chastity
MC2222	1	del Castillo	Innes
PS3333	1	White	Johnson
BU1032	2	Green	Marjorie
PC1035	1	Carson	Chervl

Курсор можно использовать в операторе SELECT так, как если бы он был обычной таблицей. Например, руководителю отдела рекламы может понадобиться упорядочить результаты просмотра в алфавитном порядке (по фамилиям авторов). Например:

SQL:

```
select *
from books
order by au_lname
```

Бухгалтеру может потребоваться другое представление. Его не интересует, кто из авторов является первым, а кто вторым. Допустим, что бухгалтера интересует только нижняя строка: на чье имя должны быть выписаны чеки и на какую сумму. Такой запрос связан с вычислением того, сколько книг было продано и по какой цене, а также какой процент от суммы полагается каждому автору:

Для данной задачи, использование курсора будет выглядеть следующим образом (ПРЕДВАРИТЕЛЬНО создайте курсор!):

Если бухгалтер воспользуется указанным ниже оператором SELECT для создания курсора под именем *royaltychecks*, то эквивалентный запрос будет иметь следующий вид:

SQL:

```
select *
from royaltychecks
```

Результаты (кто получает чек и на какую сумму) будут иметь следующий вид:

Результат:

au_lname	au_fname	Total_Income
Bennet	Abraham	4911.54
Yokomoto	Akiko	2455.36
Ringer	Albert	1421.27
Dull	Ann	4095.00
Ringer	Anne	4669.35
Gringlesby	Burt	1841.52

Наконец, рассмотрим пример со служащим головной издательской компании, который хочет выяснить положение дел с различными категориями книг в каждом из филиалов. Для этого можно воспользоваться запросом следующего вида:

SQL:

```
select pub_id, type, sum(price*ytd_sales),  
       avg(price), avg(ytd_sales)  
from titles  
group by pub_id, type
```

Однако служащему могут не нравиться все эти сложности. Что ж, можно воспользоваться куда более простым оператором:

SQL:

```
select *  
from currentinfo
```

Результат:

PUB#	TYPE	INCOME	AVG PRICE	AVG SALES
0736	business	55978.78	2.990	18722.0
0736	psychology	139319.92	13.504	1987.8
0877	NULL	NULL	NULL	NULL
0877	mod_cook	107135.22	11.49	12139.0
0877	trad_cook	249637.50	15.96	6522.0
1389	business	210036.30	17.31	4022.0
1389	popular_comp	283401.00	21.475	6437.5

Воспользовавшись этим курсором, занятой служащий может быстро увидеть,

Индивидуальное задание – разработать соответствующие подзапросы для своего варианта на языке SQL.

### ***Литература:***

1. Боуман Дж. Практическое руководство по SQL/ Дж. Боуман, С. Эмерсон, М. Мардновски, 2017. – 322 с.

### ***Инструменты:***

1. MySQLWorkbench
2. MySQL Server