Санкт-Петербургский политехнический университет

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе**

**Дисциплина**: Базы данных

**Тема**: Ознакомление с основами [SQL-DDL](http://tiger.ftk.spbstu.ru/trac/edu-db-2015/ticket/54)

Выполнил студент гр. 43501/3 Супронович П.В.

(подпись)

Руководитель Мяснов А.В.

(подпись)

Санкт-Петербург

2016

**Программа работы**

1. Самостоятельное изучение SQL-DDL
2. Создание скрипта БД в соответствии с согласованной схемой (должны присутствовать первичные и внешние ключи, ограничения на диапазоны значений). Продемонстрировать скрипт преподавателю.
3. Создайте скрипт, заполняющий все таблицы БД данными
4. Выполнение SQL-запросов, изменяющих схему созданной БД **по заданию преподавателя**. Продемонстрировать их работу преподавателю.
5. Изучите основные возможности IBExpert. Получите ER-диаграмму созданной БД с помощью **Database Designer**.
6. Автоматически сгенерируйте данные при помощи IBExpert (для трех или большего числа таблиц, не менее 100000 записей в каждой из выбранных таблиц)

**Ход работы:**

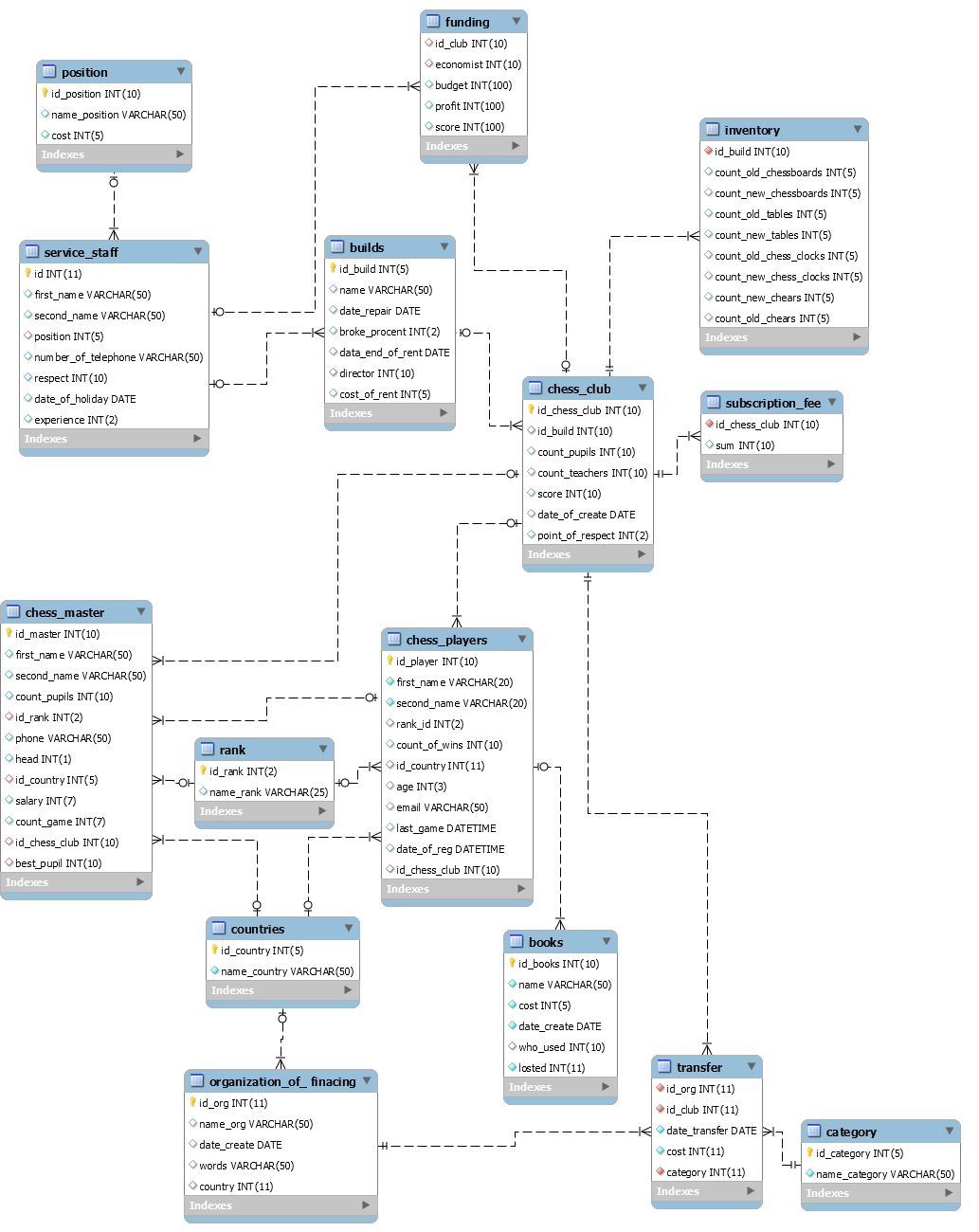


Рис. 1. ER – диаграмма базы данных шахматной индустрии

1. Для создания БД по схеме, предоставленной на рисунке 1, напишем следующий скрипт:

**SET** FOREIGN\_KEY\_CHECKS=0;

**CREATE** **DATABASE** **IF** **NOT** **EXISTS** `chessclub\_t3`;

**USE** `chessclub\_t3`;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `books` (

`id\_books` **int**(10) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер книги',

`name` **varchar**(50) **NOT** **NULL** **COMMENT** 'Наименование ',

`cost` **int**(5) **unsigned** **NOT** **NULL** **COMMENT** 'цена за книгу',

`date\_create` **date** **NOT** **NULL** **COMMENT** 'дата закупки',

`who\_used` **int**(10) **DEFAULT** '0' **COMMENT** 'кому выдана',

`losted` **int**(11) **NOT** **NULL** **COMMENT** 'утерена/не утерена',

**PRIMARY** **KEY** (`id\_books`),

**KEY** `who\_used\_book` (`who\_used`),

**CONSTRAINT** `who\_used\_book` **FOREIGN** **KEY** (`who\_used`) **REFERENCES** `chess\_players` (`id\_player`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8**;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `builds` (

`id\_build` **int**(5) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер здания',

`name` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'наименование',

`date\_repair` **date** **DEFAULT** **NULL** **COMMENT** 'дата ремонта',

`broke\_procent` **int**(2) **DEFAULT** **NULL** **COMMENT** 'процент изношенности',

`data\_end\_of\_rent` **date** **DEFAULT** **NULL** **COMMENT** 'дата окончания аренды',

`director` **int**(10) **DEFAULT** **NULL** **COMMENT** 'директор здания',

`cost\_of\_rent` **int**(5) **DEFAULT** **NULL** **COMMENT** 'арендная плата',

**PRIMARY** **KEY** (`id\_build`),

**UNIQUE** **KEY** `director` (`director`),

**CONSTRAINT** `director` **FOREIGN** **KEY** (`director`) **REFERENCES** `service\_staff` (`id`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8** **COMMENT**='здания';

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `category` (

`id\_category` **int**(5) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер категории',

`name\_category` **varchar**(50) **NOT** **NULL** **COMMENT** 'название категории',

**PRIMARY** **KEY** (`id\_category`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8**;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `chess\_club` (

`id\_chess\_club` **int**(10) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер шахматного клуба',

`id\_build` **int**(10) **DEFAULT** **NULL** **COMMENT** 'номер здания',

`count\_pupils` **int**(10) **unsigned** **DEFAULT** **NULL** **COMMENT** 'кол-во учеников',

`count\_teachers` **int**(10) **unsigned** **DEFAULT** **NULL** **COMMENT** 'кол-во учителей',

`score` **int**(10) **unsigned** **DEFAULT** **NULL** **COMMENT** 'очки за победы',

`date\_of\_create` **date** **DEFAULT** **NULL** **COMMENT** 'дата создания',

`point\_of\_respect` **int**(2) **unsigned** **DEFAULT** **NULL** **COMMENT** 'оценка клуба',

**PRIMARY** **KEY** (`id\_chess\_club`),

**UNIQUE** **KEY** `id\_build` (`id\_build`),

**CONSTRAINT** `build` **FOREIGN** **KEY** (`id\_build`) **REFERENCES** `builds` (`id\_build`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8** **COMMENT**='шахматные клубы';

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `chess\_master` (

`id\_master` **int**(10) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер мастера',

`first\_name` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'имя',

`second\_name` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'фамилия',

`count\_pupils` **int**(10) **DEFAULT** **NULL** **COMMENT** 'кол-во учеников',

`id\_rank` **int**(2) **DEFAULT** **NULL** **COMMENT** 'номер разряда',

`phone` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'телефонный номер',

`head` **int**(1) **DEFAULT** **NULL** **COMMENT** 'глава клуба(да/нет)',

`id\_country` **int**(5) **DEFAULT** **NULL** **COMMENT** 'номер страны',

`salary` **int**(7) **DEFAULT** **NULL** **COMMENT** 'заработная плата',

`count\_game` **int**(7) **DEFAULT** **NULL** **COMMENT** 'кол-во игр',

`id\_chess\_club` **int**(10) **DEFAULT** **NULL** **COMMENT** 'кол-во соревнований',

`best\_pupil` **int**(10) **DEFAULT** '0' **COMMENT** 'лучший ученик',

**PRIMARY** **KEY** (`id\_master`),

**KEY** `rank` (`id\_rank`),

**KEY** `best\_pupil` (`best\_pupil`),

**KEY** `country` (`id\_country`),

**KEY** `where\_working` (`id\_chess\_club`),

**CONSTRAINT** `best\_pupil` **FOREIGN** **KEY** (`best\_pupil`) **REFERENCES** `chess\_players` (`id\_player`),

**CONSTRAINT** `country` **FOREIGN** **KEY** (`id\_country`) **REFERENCES** `countries` (`id\_country`),

**CONSTRAINT** `rank\_master` **FOREIGN** **KEY** (`id\_rank`) **REFERENCES** `rank` (`id\_rank`),

**CONSTRAINT** `where\_working` **FOREIGN** **KEY** (`id\_chess\_club`) **REFERENCES** `chess\_club` (`id\_chess\_club`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8**;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `chess\_players` (

`id\_player` **int**(10) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'уникальный номер игрока',

`first\_name` **varchar**(20) **NOT** **NULL** **COMMENT** 'имя',

`second\_name` **varchar**(20) **NOT** **NULL** **COMMENT** 'фамилия',

`rank\_id` **int**(2) **DEFAULT** **NULL** **COMMENT** 'номер разряда',

`count\_of\_wins` **int**(10) **DEFAULT** **NULL** **COMMENT** 'кол-во побед',

`id\_country` **int**(11) **DEFAULT** **NULL** **COMMENT** 'номер страны',

`age` **int**(3) **DEFAULT** **NULL** **COMMENT** 'возраст',

`email` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'почта',

`last\_game` **datetime** **DEFAULT** **NULL** **COMMENT** 'дата послдней игры',

`date\_of\_reg` **datetime** **DEFAULT** **NULL** **COMMENT** 'дата регистрации',

`id\_chess\_club` **int**(10) **DEFAULT** **NULL** **COMMENT** 'номер шахматного клуба',

**PRIMARY** **KEY** (`id\_player`),

**UNIQUE** **KEY** `email` (`email`),

**KEY** `id\_rank` (`rank\_id`),

**KEY** `country` (`id\_country`),

**KEY** `chess\_club\_pupil` (`id\_chess\_club`),

**CONSTRAINT** `chess\_club\_pupil` **FOREIGN** **KEY** (`id\_chess\_club`) **REFERENCES** `chess\_club` (`id\_chess\_club`),

**CONSTRAINT** `country\_id` **FOREIGN** **KEY** (`id\_country`) **REFERENCES** `countries` (`id\_country`),

**CONSTRAINT** `rank\_id` **FOREIGN** **KEY** (`rank\_id`) **REFERENCES** `rank` (`id\_rank`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8**;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `countries` (

`id\_country` **int**(5) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер страны',

`name\_country` **varchar**(50) **NOT** **NULL** **COMMENT** 'наименование',

**PRIMARY** **KEY** (`id\_country`),

**UNIQUE** **KEY** `name\_country` (`name\_country`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8**;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `funding` (

`id\_club` **int**(10) **DEFAULT** **NULL** **COMMENT** 'номер клуба',

`economist` **int**(10) **DEFAULT** **NULL** **COMMENT** 'бухгалтер',

`budget` **int**(100) **DEFAULT** **NULL** **COMMENT** 'бюджет',

`profit` **int**(100) **DEFAULT** **NULL** **COMMENT** 'прибыль',

`score` **int**(100) **DEFAULT** **NULL** **COMMENT** 'расходы на содержание',

**KEY** `id\_build\_funding` (`id\_club`),

**KEY** `econom` (`economist`),

**CONSTRAINT** `econom` **FOREIGN** **KEY** (`economist`) **REFERENCES** `service\_staff` (`id`),

**CONSTRAINT** `id\_club` **FOREIGN** **KEY** (`id\_club`) **REFERENCES** `chess\_club` (`id\_chess\_club`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8** **COMMENT**='финансирование';

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `inventory` (

`id\_build` **int**(10) **NOT** **NULL** **COMMENT** 'номер здания',

`count\_old\_chessboards` **int**(5) **unsigned** **DEFAULT** '0' **COMMENT** 'кол-во старых шахматных досок',

`count\_new\_chessboards` **int**(5) **unsigned** **zerofill** **DEFAULT** '00000' **COMMENT** 'кол-во новых шахматных досок',

`count\_old\_tables` **int**(5) **unsigned** **DEFAULT** '0' **COMMENT** 'кол-во старых столов',

`count\_new\_tables` **int**(5) **unsigned** **DEFAULT** '0' **COMMENT** 'кол-во новых столов',

`count\_old\_chess\_clocks` **int**(5) **unsigned** **DEFAULT** '0' **COMMENT** 'кол-во старых шахматных часов',

`count\_new\_chess\_clocks` **int**(5) **unsigned** **DEFAULT** '0' **COMMENT** 'кол-во новых шахматных часов',

`count\_new\_chears` **int**(5) **unsigned** **DEFAULT** '0' **COMMENT** 'кол-во новых стульев',

`count\_old\_chears` **int**(5) **unsigned** **DEFAULT** '0' **COMMENT** 'кол-во старых стульев',

**KEY** `inventory` (`id\_build`),

**CONSTRAINT** `inventory` **FOREIGN** **KEY** (`id\_build`) **REFERENCES** `chess\_club` (`id\_chess\_club`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8** **COMMENT**='инвентарь(предмет считается старым, если ему больше 5 лет)';

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `organization\_of\_ finacing` (

`id\_org` **int**(11) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер организации',

`name\_org` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'имя организации',

`date\_create` **date** **DEFAULT** **NULL** **COMMENT** 'Дата создания',

`words` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'Девиз',

`country` **int**(11) **DEFAULT** **NULL** **COMMENT** 'Страна',

**PRIMARY** **KEY** (`id\_org`),

**KEY** `country\_org` (`country`),

**CONSTRAINT** `country\_org` **FOREIGN** **KEY** (`country`) **REFERENCES** `countries` (`id\_country`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8**;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `position` (

`id\_position` **int**(10) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер должности',

`name\_position` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'наименование',

`cost` **int**(5) **DEFAULT** **NULL** **COMMENT** 'заработная плата',

**PRIMARY** **KEY** (`id\_position`),

**UNIQUE** **KEY** `name\_position` (`name\_position`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8** **COMMENT**='виды должностей';

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `rank` (

`id\_rank` **int**(2) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер разряда',

`name\_rank` **varchar**(25) **DEFAULT** **NULL** **COMMENT** 'наименование разряда',

**PRIMARY** **KEY** (`id\_rank`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8** **COMMENT**='разряд по шахматам';

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `service\_staff` (

`id` **int**(11) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер сотрудника',

`first\_name` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'имя',

`second\_name` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'фамилия',

`position` **int**(5) **DEFAULT** **NULL** **COMMENT** 'номер должности',

`number\_of\_telephone` **varchar**(50) **DEFAULT** **NULL** **COMMENT** 'номер телефона',

`respect` **int**(10) **DEFAULT** **NULL** **COMMENT** 'репутация',

`date\_of\_holiday` **date** **DEFAULT** **NULL** **COMMENT** 'дата последнего отпуска',

`experience` **int**(2) **DEFAULT** **NULL** **COMMENT** 'стаж',

**PRIMARY** **KEY** (`id`),

**KEY** `position` (`position`),

**CONSTRAINT** `position` **FOREIGN** **KEY** (`position`) **REFERENCES** `position` (`id\_position`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8** **COMMENT**='обслуживающий персонал';

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `subscription\_fee` (

`id\_chess\_club` **int**(10) **NOT** **NULL** **COMMENT** 'номер шахматного клуба',

`sum` **int**(10) **unsigned** **DEFAULT** **NULL** **COMMENT** 'цена',

**KEY** `subscription fee` (`id\_chess\_club`),

**CONSTRAINT** `subscription fee` **FOREIGN** **KEY** (`id\_chess\_club`) **REFERENCES** `chess\_club` (`id\_chess\_club`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8** **COMMENT**='абонентская плата';

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `transfer` (

`id\_org` **int**(11) **NOT** **NULL** **COMMENT** 'номер организации',

`id\_club` **int**(11) **NOT** **NULL** **COMMENT** 'номер спансируемого клуба',

`date\_transfer` **date** **NOT** **NULL** **COMMENT** 'дата перевода',

`cost` **int**(11) **NOT** **NULL** **COMMENT** 'сколько',

`category` **int**(11) **NOT** **NULL** **COMMENT** 'категория перевода',

**KEY** `link\_or` (`id\_org`),

**KEY** `link\_club` (`id\_club`),

**KEY** `link\_category` (`category`),

**CONSTRAINT** `link\_category` **FOREIGN** **KEY** (`category`) **REFERENCES** `category` (`id\_category`),

**CONSTRAINT** `link\_club` **FOREIGN** **KEY** (`id\_club`) **REFERENCES** `chess\_club` (`id\_chess\_club`),

**CONSTRAINT** `link\_or` **FOREIGN** **KEY** (`id\_org`) **REFERENCES** `organization\_of\_ finacing` (`id\_org`)

) **ENGINE**=**InnoDB** **DEFAULT** **CHARSET**=**utf8** **COMMENT**='список организаций, занимающихся финансированием';

**SET** FOREIGN\_KEY\_CHECKS=1;

1. Создадим скрипт, заполняющий все таблицы БД данными.

Был написан скрипт на php, генерирующий случайные данные для различных таблиц и посылающий в базу данных. Код скрипта прилагается к отчету. Благодаря сгенерированным данным становится возможно тестировать имеющиеся запросы и оптимизировать их в случае несоответствия требованиям производительности.

БД была приведена к нормальной форме. Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц).  
Основные критерии первой нормальной формы (1NF):

1. Все строки должны быть различными.
2. Все элементы внутри ячеек должны быть атомарными (не списками). Другими словами, элемент является атомарным, если его нельзя разделить на части, которые могут использовать в таблице независимо друг от друга.

Основные критерии второй нормальной формы (2NF):

1. Таблица должна находиться в первой нормальной форме.
2. Любое её поле, не входящее в состав первичного ключа, функционально полно зависит от первичного ключа.

Основные критерии третьей нормальной формы (3NF):

1. Таблица находится во второй нормальной форме.
2. Любой её не ключевой атрибут функционально зависит только от первичного ключа.

Выводы

Язык SQL-DDL определяет структуру и ограничения целостности баз данных. Сюда относятся команды создания и удаления баз данных; создания, изменения и удаления таблиц; управления пользователями и т.д.

Язык SQL-DDL является декларативным языком, чем отличается от ранее изученных языков. Используя декларативный язык, необходимо описать результат, а не метод его получения, как это делается в процедурных языках, например в С/С++, Pascal и т.д.

**Индивидуальное задание**

***Задание:***

Модифицировать схему для удовлетворения следующим требованиям:

1. Ввести учет закупок имущества.
2. Ввести учет амортизации и списания имущества.

Для выполнения поставленных задач было решено удалить таблицу инвентарь и добавить две таблицы: покупки и амортизация.

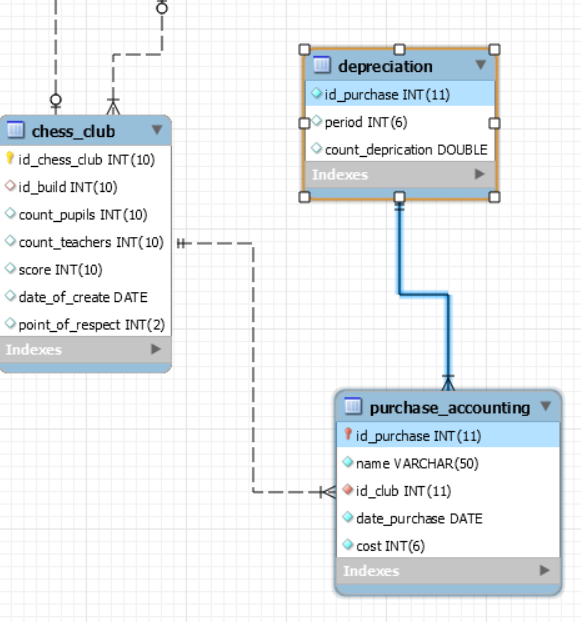


Рис. 1 – часть Er-диаграммы

С помощью таблицы depreciation будет вестись учет амортизации, а таблица purchase\_aссounting - позволяет вести учет закупок. Для добавления этих таблиц был написан следующий код:

**CREATE** **TABLE** `purchase\_accounting` (

`id\_purchase` **INT**(11) **NOT** **NULL** **AUTO\_INCREMENT** **COMMENT** 'номер покупки',

`name` **VARCHAR**(50) **NOT** **NULL** **DEFAULT** '0' **COMMENT** 'имя покупки',

`id\_club` **INT**(11) **NOT** **NULL** **DEFAULT** '0' **COMMENT** 'клуб в который купили',

`date\_purchase` **DATE** **NOT** **NULL** **COMMENT** 'дата покупки',

`cost` **INT**(6) **UNSIGNED** **NOT** **NULL** **DEFAULT** '0' **COMMENT** 'цена покупки',

**PRIMARY** **KEY** (`id\_purchase`),

**INDEX** `FK\_purchase\_accounting\_chess\_club` (`id\_club`),

**CONSTRAINT** `FK\_purchase\_accounting\_chess\_club` **FOREIGN** **KEY** (`id\_club`) **REFERENCES** `chess\_club` (`id\_chess\_club`),

**CONSTRAINT** `FK\_purchase\_accounting\_depreciation` **FOREIGN** **KEY** (`id\_purchase`) **REFERENCES** `depreciation` (`id\_purchase`)

);

**CREATE** **TABLE** `depreciation` (

`id\_purchase` **INT**(11) **NOT** **NULL** **COMMENT** 'номер покупки',

`period` **INT**(6) **NULL** **DEFAULT** **NULL** **COMMENT** 'кол-во дней до 0 аморт.',

`count\_deprication` **DOUBLE** **NULL** **DEFAULT** **NULL** **COMMENT** 'величин. аморт.',

**UNIQUE** **INDEX** `id\_purchase` (`id\_purchase`)

);

В результате ER диаграмма выглядит следующим образом:

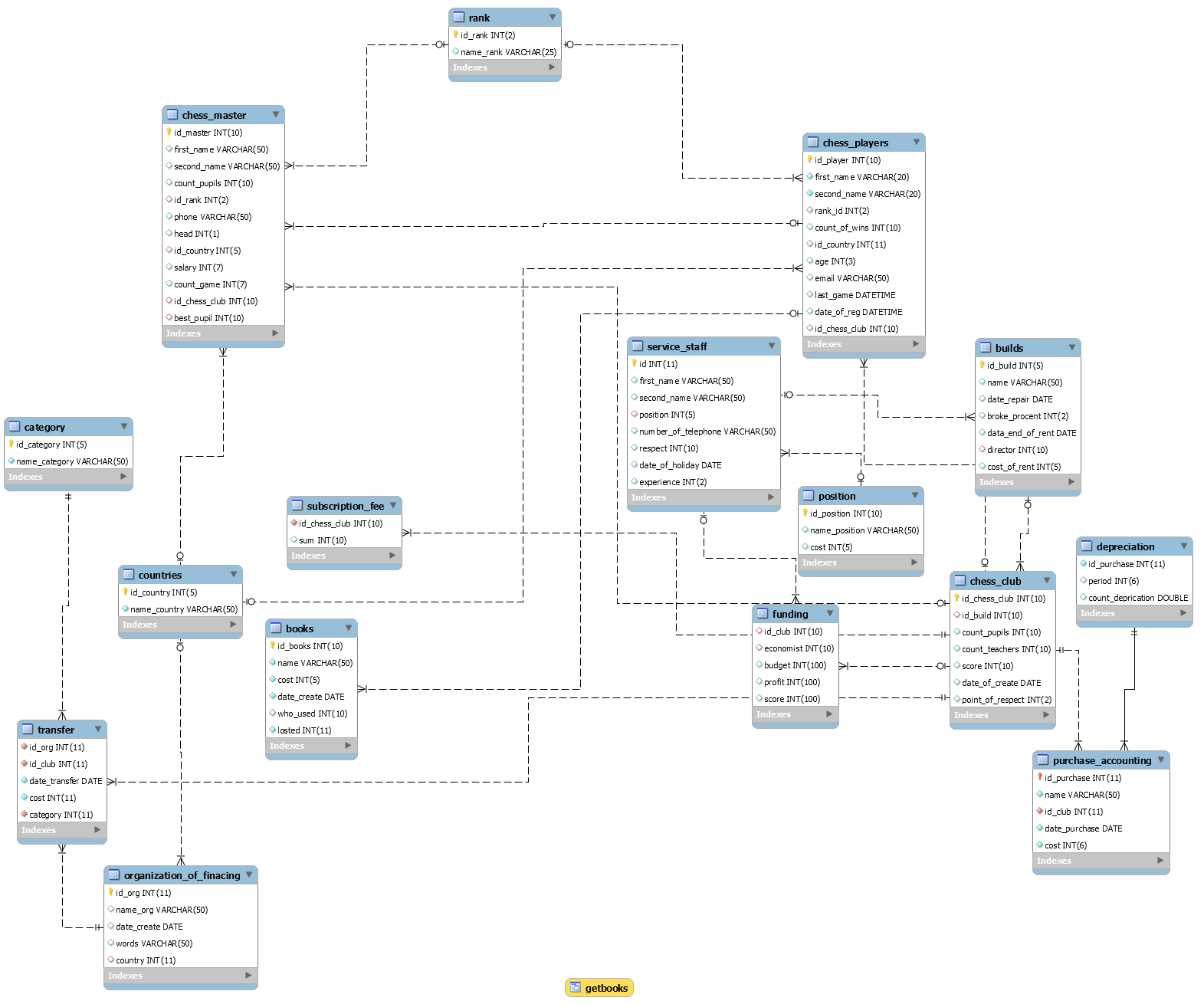


Рис. 2 – итоговая Er-диаграмма

**Итог:**

Задание выполнено, база данных модифицирована.

**Дополнено:**

Необходимо создать таблицу «списывание», которая содержала бы в себе информацию о списанном имуществе: дату списания, ссылку на предмет.

**CREATE** **TABLE** `cheating` (

`id\_cheating` **INT**(11) **NOT** **NULL** **AUTO\_INCREMENT**,

`id\_purchase` **INT**(11) **NOT** **NULL** **DEFAULT** '0',

`date` **DATE** **NOT** **NULL**,

**PRIMARY** **KEY** (`id\_cheating`),

**INDEX** `id\_purchase` (`id\_purchase`),

**CONSTRAINT** `id\_purchase` **FOREIGN** **KEY** (`id\_purchase`) **REFERENCES** `purchase\_accounting` (`id\_purchase`)

)

**COMMENT**='таблица со списанными предметами'

**COLLATE**='utf8\_general\_ci'

**ENGINE**=**InnoDB**

Er–диаграмма выглядит следующим образом:

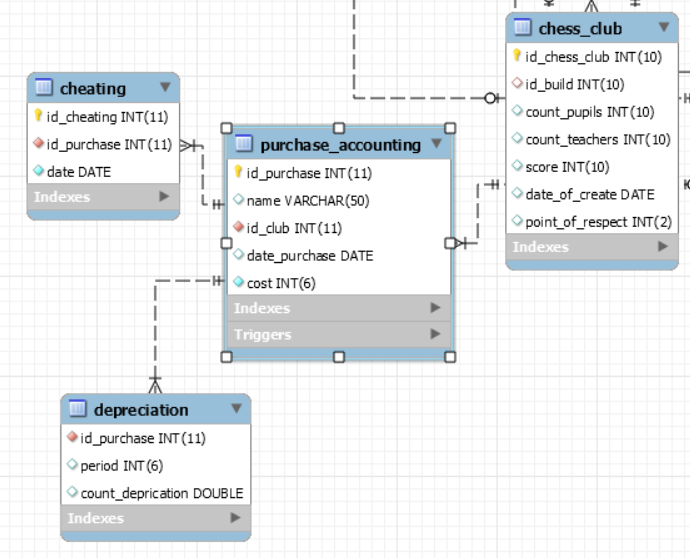


Рис. 3 – часть Er-диаграммы

Есть таблица предметов, которая связана внешними ключами с таблицами амортизации и списывания.