МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский авиационный институт (национальный исследовательский университет)»

Ступинский филиал МАИ

Кафедра «Моделирование систем и информационные технологии»

**КУРСОВАЯ РАБОТА**

по дисциплине «Базы данных»

**База данных «Orna»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | | | *Данилин А.С.* |  |
| Группа | | | *ТСЗ-301Бк-18* |  |
| Руководитель | | | *Лезжова А. М.* |  |
| Оценка |  | Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_г. | | |

**Ступино 2020**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский авиационный институт (национальный исследовательский университет)»

Ступинский филиал МАИ

Кафедра «Моделирование систем и информационные технологии»

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. М. Мамонов

(И.О.Фамилия)

« \_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_ г.

**З А Д А Н И Е**

на курсовой проект по дисциплине

«Базы данных»

Студент ФИО

(№ группы, Ф. И. О.)

Тема База данных вселенной Трансформеров

Перечень вопросов, подлежащих разработке в курсовой работе

1. Проектирование БД.
2. Разработка схемы реляционной базы данных, определение ключей, связей, типов данных, описание структуры БД.
3. Разработка инструкций CREATE TABLE для создания таблиц в среде Microsoft SQL Server.
4. Выполнение нормализации и расчёта объема БД.
5. Выполнение запросов к БД.

Рекомендуемая литература

1. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / СПб.: КОРОНА принт, 2004, 736 с.
2. Григорьев Ю.А., Ревунков Г.И. Банки данных: Учеб. для вузов / М.: Изд-во МГТУ им. Баумана Н.Э. 2002, 320с.
3. Махмутова М.В. Теория и практика разработки баз данных, 2017 г. - коллекция "Информатика - Издательство ФЛИНТА". Режим доступа: https://e.lanbook.com/book/104917
4. Храпченко М.В. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для проектирования информационных систем : учеб. пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — М. : ИД «ФОРУМ» : ИНФРА-М, 2019. — 368 с. Режим доступа: http://znanium.com/catalog/product/1001370
5. Гаскаров Д.В. Интеллектуальные информационные системы. Учеб. для вузов. - М.: Высш.шк., 2003. - 431 с.

Задание выдано «\_\_\_»\_\_\_\_\_\_\_\_\_20\_\_ г.

Руководитель

(Ф. И. О., должность, подпись)

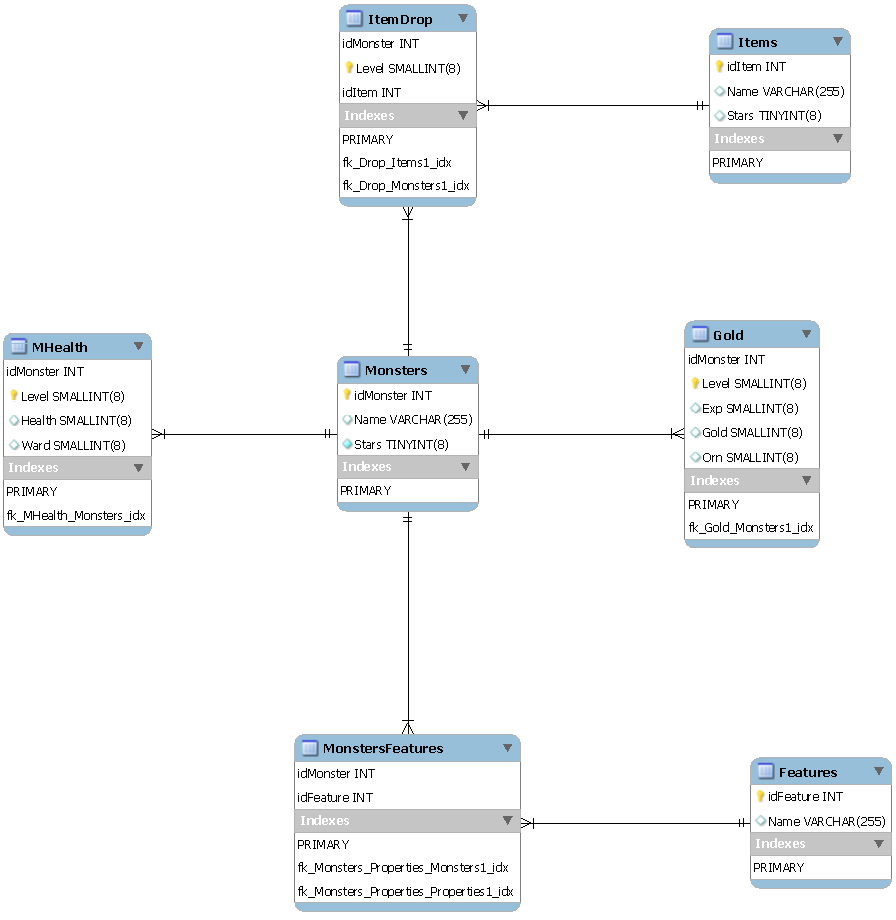
Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

**Постановка задачи**

В представленной БД поставлена задача систематизировать сведения о монстрах в игре «Orna» - количество опыта, получаемые предметы, слабые и сильные стороны. В БД будут использоваться разные типы данных и различные отношения между таблицами.

**Схема базы данных**

****

**Описание таблиц**

Во всех столбцах всех таблиц значения NULL запрещены. Теоретически. Так как данные получаются опытным, столбцы помеченные значком «\*» могут содержать NULL.

**1)** **Монстры (основная таблица)**

Таблица содержит следующие столбцы:

* idMonster – первичный ключ, уникальный идентификатор;
* Name – Наименование монстра видимое на карте;
* Stars – Условный уровень силы монстра;

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Размер, байт** |
| idMonster | Int | 4 |
| Name | Varchar(255) | 255 |
| Stars | Tinyint | 1 |

Итого одна запись в данной таблице будет занимать 260 байт.

**2) Gold**

Определяет количество опыта, золота и орн (спец. валюта применяется для улучшения оружия, брони, приобретения питомцев),для каждого монстра, каждого уровня.

Содержит следующие столбцы:

* idMonster – внешний ключ, первичный ключ;
* Level – уровень монстра, первичный ключ;
* Exp – Количество получаемого опыта;
* Gold – Количество получаемого золота;
* Orn – Количество получаемого орн.

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Размер, байт** |
| idMonster | Int | 4 |
| Level | Smallint | 2 |
| Exp | Smallint | 2 |
| Gold | Smallint | 2 |
| Orn | Smallint | 2 |

Итого одна запись в таблице занимает 12 байт.

**3) MHealth**

Определяет количество здоровья и шита, для каждого монстра, каждого уровня.

Содержит следующие столбцы:

* idMonster – внешний ключ, первичный ключ;
* Level – уровень монстра, первичный ключ;
* Health – Количество здоровья монстра;
* Ward\* – Количество шита монстра.

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Размер, байт** |
| idMonster | Int | 4 |
| Level | smallint | 2 |
| Health | smallint | 2 |
| Ward\* | smallint | 2 |

Итого одна запись в таблице занимает 10 байт.

**4) ItemsDrop**

Список выпадающих с монстров предметов, для каждого монстра, каждого уровня.

Содержит следующие столбцы:

* idMonster – внешний ключ, первичный ключ;
* Level – уровень монстра, первичный ключ;
* idItem – внешний ключ, идентификатор предмета.

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Размер, байт** |
| idMonster | Int | 4 |
| Level | smallint | 2 |
| idItem | Int | 4 |

Итого одна запись в таблице занимает 10 байта.

**5) Items**

Список существующих в игре предметов.

Содержит следующие столбцы:

* idItem – первичный ключ, уникальный идентификатор;
* Name – Наименование предмета;
* Stars – Условный уровень силы предмета*.*

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Размер, байт** |
| idItem | Int | 4 |
| Name | Varchar(255) | 255 |
| Stars | Tinyint | 1 |

Итого одна запись в таблице занимает 260 байт.

**6) MonstersFeatures**

Список особенностей мостров, например уязвимости и сопротивлениея.

Содержит следующие столбцы:

* idMonster – часть составного ключа;
* idFeatures– внешний ключ к таблице Features, часть составного ключа.

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Размер, байт** |
| idMonster | Int | 4 |
| idFeatures | Int | 4 |

Итого одна запись в таблице занимает 8 байт.

**7) Features**

Список существующих особенностей монстров.

Содержит следующие столбцы:

* idFeature – первичный ключ, уникальный идентификатор;
* Name – наименование предмета;

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Размер, байт** |
| idFeature | Int | 4 |
| Name | Varchar(255) | 255 |

Итого одна запись в таблице занимает 259 байта.

**Листинг SQL запроса для пересоздания БД**

**Листинг запросов SQL для заполнения некоторых таблиц**

**Листинг запросов с выборкой**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278 | -- MySQL Workbench Synchronization  -- Generated: 2020-10-03 18:02  -- Model: Orna  -- Version: 1.0  -- Project: SF MAI Databases  -- Author: Danilin Aleksander  **-- Создаем схему**  **CREATE** **SCHEMA** IF **NOT** **EXISTS** `Orna` **DEFAULT** CHARACTER **SET** cp1251 ;  **-- Удаляем структуру**  **DROP** **TABLE** IF **EXISTS** `Orna`.`itemsdrop`;  **DROP** **TABLE** IF **EXISTS** `Orna`.`mhealth`;  **DROP** **TABLE** IF **EXISTS** `Orna`.`monstersfeatures`;  **DROP** **TABLE** IF **EXISTS** `Orna`.`gold`;  **DROP** **TABLE** IF **EXISTS** `Orna`.`features`;  **DROP** **TABLE** IF **EXISTS** `Orna`.`items`;  **DROP** **TABLE** IF **EXISTS** `Orna`.`monsters`;  **DROP** **PROCEDURE** IF **EXISTS** `Orna`.`sp\_Items\_Drops`;  **DROP** **PROCEDURE** IF **EXISTS** `Orna`.`sp\_Monsters\_Drops`;  **DROP** **PROCEDURE** IF **EXISTS** `Orna`.`sp\_Monsters\_Features`;  **DROP** **PROCEDURE** IF **EXISTS** `Orna`.`sp\_Monsters\_Gold`;  **DROP** **PROCEDURE** IF **EXISTS** `Orna`.`sp\_Monsters\_Health`;  **-- Создаем таблицы**  **CREATE** **TABLE** IF **NOT** **EXISTS** `Orna`.`Monsters` (  `idMonster` INT(**10**) UNSIGNED **NOT** **NULL** AUTO\_INCREMENT,  `Name` VARCHAR(**255**) **NULL** **DEFAULT** **NULL**,  `Stars` TINYINT(**8**) **NOT** **NULL**,  **PRIMARY** **KEY** (`idMonster`))  ENGINE = InnoDB  **DEFAULT** CHARACTER **SET** = cp1251;  **CREATE** **TABLE** IF **NOT** **EXISTS** `Orna`.`MHealth` (  `idMonster` INT(**10**) UNSIGNED **NOT** **NULL**,  `**Level**` SMALLINT(**8**) **NOT** **NULL**,  `Health` SMALLINT(**8**) **NULL** **DEFAULT** **NULL**,  `Ward` SMALLINT(**8**) **NULL** **DEFAULT** **NULL**,  **PRIMARY** **KEY** (`**Level**`, `idMonster`),  **INDEX** `fk\_MHealth\_Monsters\_idx` (`idMonster` **ASC**),  **CONSTRAINT** `fk\_MHealth\_Monsters`  **FOREIGN** **KEY** (`idMonster`)  **REFERENCES** `Orna`.`Monsters` (`idMonster`)  **ON** **DELETE** **NO** ACTION  **ON** **UPDATE** **NO** ACTION)  ENGINE = InnoDB;  **CREATE** **TABLE** IF **NOT** **EXISTS** `Orna`.`Gold` (  `idMonster` INT(**10**) UNSIGNED **NOT** **NULL**,  `**Level**` SMALLINT(**8**) **NOT** **NULL**,  `Exp` SMALLINT(**8**) **NULL** **DEFAULT** **NULL**,  `Gold` SMALLINT(**8**) **NULL** **DEFAULT** **NULL**,  `Orn` SMALLINT(**8**) **NULL** **DEFAULT** **NULL**,  **PRIMARY** **KEY** (`idMonster`, `**Level**`),  **INDEX** `fk\_Gold\_Monsters1\_idx` (`idMonster` **ASC**),  **CONSTRAINT** `fk\_Gold\_Monsters1`  **FOREIGN** **KEY** (`idMonster`)  **REFERENCES** `Orna`.`Monsters` (`idMonster`)  **ON** **DELETE** **NO** ACTION  **ON** **UPDATE** **NO** ACTION)  ENGINE = InnoDB;  **CREATE** **TABLE** IF **NOT** **EXISTS** `Orna`.`Items` (  `idItem` INT(**10**) UNSIGNED **NOT** **NULL** AUTO\_INCREMENT,  `Name` VARCHAR(**255**) **NULL** **DEFAULT** **NULL**,  `Stars` TINYINT(**8**) **NULL** **DEFAULT** **NULL**,  **PRIMARY** **KEY** (`idItem`))  ENGINE = InnoDB  **DEFAULT** CHARACTER **SET** = cp1251;  **CREATE** **TABLE** IF **NOT** **EXISTS** `Orna`.`ItemsDrop` (  `idMonster` INT(**10**) UNSIGNED **NOT** **NULL**,  `**Level**` SMALLINT(**8**) **NOT** **NULL**,  `idItem` INT(**10**) UNSIGNED **NOT** **NULL**,  **PRIMARY** **KEY** (`**Level**`, `idMonster`, `idItem`),  **INDEX** `fk\_Drop\_Items1\_idx` (`idItem` **ASC**),  **INDEX** `fk\_Drop\_Monsters1\_idx` (`idMonster` **ASC**),  **CONSTRAINT** `fk\_Drop\_Items1`  **FOREIGN** **KEY** (`idItem`)  **REFERENCES** `Orna`.`Items` (`idItem`)  **ON** **DELETE** **NO** ACTION  **ON** **UPDATE** **NO** ACTION,  **CONSTRAINT** `fk\_Drop\_Monsters1`  **FOREIGN** **KEY** (`idMonster`)  **REFERENCES** `Orna`.`Monsters` (`idMonster`)  **ON** **DELETE** **NO** ACTION  **ON** **UPDATE** **NO** ACTION)  ENGINE = InnoDB;  **CREATE** **TABLE** IF **NOT** **EXISTS** `Orna`.`Features` (  `idFeature` INT(**10**) UNSIGNED **NOT** **NULL** AUTO\_INCREMENT,  `Name` VARCHAR(**255**) **NULL** **DEFAULT** **NULL**,  **PRIMARY** **KEY** (`idFeature`))  ENGINE = InnoDB  **DEFAULT** CHARACTER **SET** = cp1251;  **CREATE** **TABLE** IF **NOT** **EXISTS** `Orna`.`MonstersFeatures` (  `idMonster` INT(**10**) UNSIGNED **NOT** **NULL**,  `idFeature` INT(**10**) UNSIGNED **NOT** **NULL**,  **PRIMARY** **KEY** (`idMonster`, `idFeature`),  **INDEX** `fk\_Monsters\_Features\_Monsters1\_idx` (`idMonster` **ASC**),  **INDEX** `fk\_Monsters\_Features\_Features1\_idx` (`idFeature` **ASC**),  **CONSTRAINT** `fk\_Monsters\_Features\_Monsters1`  **FOREIGN** **KEY** (`idMonster`)  **REFERENCES** `Orna`.`Monsters` (`idMonster`)  **ON** **DELETE** **NO** ACTION  **ON** **UPDATE** **NO** ACTION,  **CONSTRAINT** `fk\_Monsters\_Features\_Features1`  **FOREIGN** **KEY** (`idFeature`)  **REFERENCES** `Orna`.`Features` (`idFeature`)  **ON** **DELETE** **NO** ACTION  **ON** **UPDATE** **NO** ACTION)  ENGINE = InnoDB;  **-- Создаем процедуры**  **DELIMITER** $$  USE `Orna`$$  /\* Здоровье выбранного монстра по уровням \*/  **CREATE** **PROCEDURE** `sp\_Monsters\_Health` (prm\_MonsterID INT)  **BEGIN**  **SELECT** m.Name **AS** name  , m.Stars **AS** Stars  , h.**Level** **AS** **Level**  , h.Health **AS** Health  , h.Ward **AS** Ward  **FROM** Monsters **AS** m  **JOIN** MHealth **AS** h  **ON** h.idMonster = m.idMonster  **WHERE** m.idMonster = prm\_MonsterID  **ORDER** **BY** h.**Level**;  **END**$$  /\* Список вещей выпадающих с монстра \*/  **CREATE** **PROCEDURE** `sp\_Monsters\_Drops` (prm\_MonsterID INT)  **BEGIN**  **SELECT** m.Name **AS** Name  , m.Stars **AS** Stars  , REPEAT('\*', m.Stars) **AS** StrStars  , GROUP\_CONCAT(i.Name SEPARATOR ', ') **AS** Item  **FROM** Monsters **AS** m  **JOIN** ItemsDrop **AS** d  **ON** d.idMonster = m.idMonster  **JOIN** Items **AS** i  **ON** d.idItem = i.idItem  **WHERE** m.idMonster = prm\_MonsterID;  **END**$$  /\* Количество золота и опыта за монстра по уровням \*/  **CREATE** **PROCEDURE** `sp\_Monsters\_Gold` (prm\_MonsterID INT)  **BEGIN**  **SELECT** m.Name **AS** Name  , **g**.**Level** **AS** **Level**  , m.Stars **AS** Stars  , REPEAT('\*', m.Stars) **AS** StrStars  , **g**.Exp **AS** Exp  , **g**.Gold **AS** Gold  , **g**.Orn **AS** Orn  **FROM** Monsters **AS** m  **JOIN** Gold **AS** **g**  **ON** **g**.idMonster = m.idMonster  **WHERE** m.idMonster = prm\_MonsterID  **ORDER** **BY** **g**.**Level**;  **END**$$  /\* Список монстров с которых выпадает предмет \*/  **CREATE** **PROCEDURE** `sp\_Items\_Drops` (prm\_ItemID INT)  **BEGIN**  **SELECT** i.Name **AS** Item  , i.Stars **AS** Item\_stars  , REPEAT('\*', i.Stars) **AS** StrStars  , GROUP\_CONCAT(m.Name SEPARATOR ', ') **AS** Monsters  **FROM** Items **AS** i  **JOIN** ItemsDrop **AS** d  **ON** d.idItem = i.idItem  **JOIN** Monsters **AS** m  **ON** m.idMonster = d.idMonster  **WHERE** i.idItem = prm\_ItemID;  **END**$$  /\* Список особенностей монстров \*/  **CREATE** **PROCEDURE** `sp\_Monsters\_Features` ()  **BEGIN**  **SELECT** f.Name **AS** Feature  , m.Name **AS** Monter  **FROM** Monsters **AS** m  **JOIN** Monsters\_Features **AS** mp  **ON** mp.idMonter = m.idMonster  **JOIN** Features **AS** f  **ON** f.idFeature = mp.idFeature  **GROUP** **BY** m.idMonster;  **END**$$  **DELIMITER** ;  **-- Заполняем таблицы**  -- Базовые  **INSERT** **INTO** `Orna`.`monsters`  (`idMonster`, `Name`, `Stars`)  **VALUES**  (**1**, 'Паук', **1**),  (**2**, 'Мустанг', **2**),  (**3**, 'Красный слизень', **1**),  (**4**, 'Ястреб', **2**),  (**5**, 'Драконит-маг', **2**);  **INSERT** **INTO** `Orna`.`items`  (`idItem`, `Name`, `Stars`)  **VALUES**  (**1**, 'Малое зелье маны', **1**),  (**2**, 'Зелье лечения', **2**),  (**3**, 'Кожа', **1**),  (**4**, 'Малое зелье маны', **1**),  (**5**, 'Малый элексир', **2**),  (**6**, 'Зелье маны', **2**),  (**7**, 'Книга чудищ', **2**),  (**8**, 'Клобук драконита', **2**),  (**9**, 'Роба драконита', **2**),  (**10**, 'Посох драконита', **2**);  **INSERT** **INTO** `Orna`.`features`  (`idFeature`, `Name`)  **VALUES**  (**null**, **null**);  -- Производные  **INSERT** **INTO** `Orna`.`gold`  (`idMonster`, `**Level**`, `Exp`, `Gold`, `Orn`)  **VALUES**  (**1**, **3**, **1**, **6**, **7**),  (**2**, **38**, **1179**, **85**, **13**),  (**2**, **40**, **1267**, **108**, **11**),  (**2**, **47**, **1451**, **177**, **13**),  (**3**, **4**, **107**, **7**, **6**),  (**4**, **42**, **1434**, **96**, **13**),  (**5**, **39**, **1335**, **87**, **10**);  **INSERT** **INTO** `Orna`.`itemsdrop`  (`idMonster`, `**Level**`, `idItem`)  **VALUES**  (**1**, **3**, **1**),  (**2**, **40**, **1**),  (**2**, **38**, **2**),  (**2**, **47**, **2**),  (**2**, **47**, **3**),  (**2**, **40**, **4**),  (**2**, **38**, **5**),  (**5**, **39**, **6**),  (**5**, **39**, **7**),  (**5**, **39**, **8**),  (**5**, **39**, **9**),  (**5**, **39**, **10**);  **INSERT** **INTO** `Orna`.`mhealth`  (`idMonster`, `**Level**`, `Health`, `Ward`)  **VALUES**  (**1**, **3**, **29**, **NULL**),  (**3**, **4**, **16**, **NULL**),  (**2**, **38**, **355**, **NULL**),  (**5**, **39**, **342**, **NULL**),  (**2**, **40**, **350**, **NULL**),  (**2**, **42**, **365**, **NULL**),  (**4**, **42**, **365**, **NULL**),  (**2**, **47**, **402**, **91**);  /\*INSERT INTO `Orna`.`monstersfeatures`  (`idMonster`, `idFeature`)  VALUES  (NULL, NULL);\*/ |

**Расчет объема БД**

|  |  |  |  |
| --- | --- | --- | --- |
| **Таблица** | **Размер записи, байт** | **Кол-во записей** | **Объем таблицы, байт** |
| Monsters | 260 | 5 | 1300 |
| Gold | 12 | 7 | 84 |
| MHealth | 10 | 12 | 120 |
| Items | 260 | 10 | 2600 |
| ItemsDrop | 10 | 8 | 80 |
| Features | 259 | 0 | 0 |
| MonstersFeatures | 8 | 0 | 0 |
| **Объем базы данных, байт** | | | **4184** |