

Федеральное государственное автономное образовательное учреждение высшего образования
"Национальный Исследовательский Университет ИТМО"
Мегафакультет Компьютерных Технологий и Управления
Факультет Программной Инженерии и Компьютерной Техники



Курсовая работа этап 3
по дисциплине
'Информационные системы и базы данных'

Выполнили Студенты группы Р33102

Лапин Алексей Александрович

Юнусов Роман Ильдарович

Преподаватель:

Сагайдак Алина Алексеевна

г. Санкт-Петербург
2023г.

Содержание

1	Текст задания.	3
2	Описание предметной области.	3
3	Даталогическая модель.	4
4	Создать необходимые объекты базы данных.	4
4.1	Создание таблиц:	4
5	Заполнить созданные таблицы тестовыми данными.	7
6	Сделать скрипты для:	13
6.1	создания/удаления объектов базы данных;	13
6.2	заполнения/удаления созданных таблиц;	13
7	Добавить в базу данных триггеры для обеспечения комплексных ограничений целостности.	14
8	Реализовать функции и процедуры на основе описания бизнес-процессов (из этапа №1).	15
9	Произвести анализ использования созданной базы данных:	17
10	Создать индексы и доказать, что они полезны для вашей базы данных:	17

1 Текст задания.

Реализовать даталогическую модель в реляционной СУБД PostgreSQL:

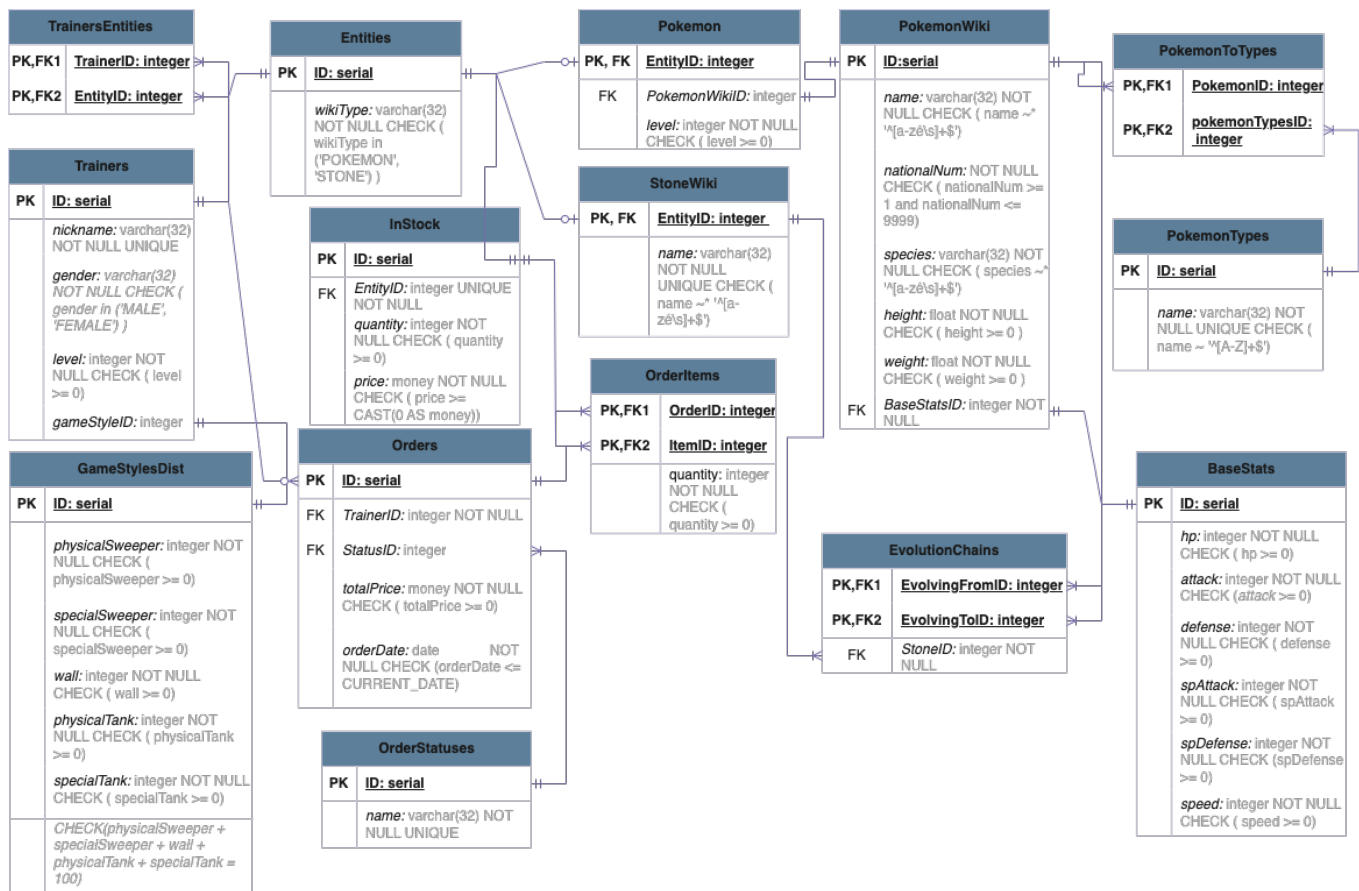
- Создать необходимые объекты базы данных.
- Заполнить созданные таблицы тестовыми данными.
- Сделать скрипты для:
 - создания/удаления объектов базы данных;
 - заполнения/удаления созданных таблиц.
- Обеспечить целостность данных при помощи средств языка DDL.
- Добавить в базу данных триггеры для обеспечения комплексных ограничений целостности.
- Реализовать функции и процедуры на основе описания бизнес-процессов (из этапа №1).
- Произвести анализ использования созданной базы данных:
 - выявить наиболее часто используемые запросы к объектам базы данных;
 - результаты представить в виде текстового описания.
- Создать индексы и доказать, что они полезны для вашей базы данных:
 - доказательство должно быть приведено в виде текстового описания.

2 Описание предметной области.

Магазин покемонов.

Существуют тренеры и покемоны. В нашем магазине продаются покемоны и камни эволюции. Наш магазин умеет предлагать к покупке покемонов соответствующие уровню и стилю игры тренера. Некоторые из покемонов могут эволюционировать в других покемонов при применении подходящего камня эволюции.

3 Дatalogическая модель.



4 Создать необходимые объекты базы данных.

4.1 Создание таблиц:

```

1 CREATE TABLE IF NOT EXISTS entities(
2     id                serial          PRIMARY KEY,
3     wikiType          varchar(32)     NOT NULL CHECK ( wikiType in
4         ('POKEMON', 'STONE') )
5 );
6
7 CREATE TABLE IF NOT EXISTS baseStats(
8     id                serial          PRIMARY KEY,
9     hp                integer         NOT NULL CHECK ( hp >= 0),
10    attack             integer         NOT NULL CHECK ( attack >= 0),
11    defense            integer         NOT NULL CHECK ( defense >= 0),
12    spAttack           integer         NOT NULL CHECK ( spAttack >= 0),
13    spDefense          integer         NOT NULL CHECK ( spDefense >= 0),
14    speed              integer         NOT NULL CHECK ( speed >= 0)
15 );
16
17 CREATE TABLE IF NOT EXISTS pokemonWiki(
18     id                serial          PRIMARY KEY,
19     name              varchar(32)     NOT NULL CHECK ( name ~* '[a-zé\s]+$'),

```

```

19     nationalNum integer          NOT NULL CHECK ( nationalNum >= 1 and
        nationalNum <= 9999),
20     species     varchar(32)      NOT NULL CHECK ( species ~*
        '~[a-zé\s]+$'),
21     height      float            NOT NULL CHECK ( height >= 0 ),
22     weight      float            NOT NULL CHECK ( weight >= 0 ),
23     baseStatsID integer          NOT NULL REFERENCES baseStats(id) ON
        DELETE CASCADE
24 );
25
26 CREATE TABLE IF NOT EXISTS pokemon(
27     entityID     integer          PRIMARY KEY REFERENCES entities(id)
        ON DELETE CASCADE,
28     pokemonWikiID integer         NOT NULL REFERENCES pokemonWiki(id)
        ON DELETE CASCADE,
29     level        integer          NOT NULL CHECK ( level >= 0)
30 );
31
32 CREATE TABLE IF NOT EXISTS pokemonTypes(
33     id           serial           PRIMARY KEY,
34     name         varchar(32)      NOT NULL UNIQUE CHECK ( name ~
        '~[A-Z]+$')
35 );
36
37 CREATE TABLE IF NOT EXISTS pokemonToTypes(
38     pokemonID     integer         REFERENCES pokemonWiki(id) ON DELETE
        CASCADE,
39     pokemonTypesID integer        REFERENCES pokemonTypes(id) ON DELETE
        CASCADE,
40     PRIMARY KEY (pokemonID, pokemonTypesID)
41 );
42
43 CREATE TABLE IF NOT EXISTS stoneWiki(
44     entityID     integer          PRIMARY KEY REFERENCES entities(id)
        ON DELETE CASCADE,
45     name         varchar(32)      UNIQUE NOT NULL CHECK ( name ~*
        '~[a-zé\s]+$')
46 );
47
48 CREATE TABLE IF NOT EXISTS evolutionChains(
49     EvolvingFromID integer         REFERENCES pokemonWiki(id) ON DELETE
        CASCADE,
50     EvolvingToID   integer         REFERENCES pokemonWiki(id) ON DELETE
        CASCADE,
51     stoneID        integer         NOT NULL REFERENCES stoneWiki(entityID)
        ON DELETE CASCADE,
52     PRIMARY KEY (EvolvingFromID, EvolvingToID)
53 );
54
55
56

```

```

57 CREATE TABLE IF NOT EXISTS inStock(
58     id          serial          PRIMARY KEY,
59     entityID integer          NOT NULL UNIQUE REFERENCES entities(id) ON
        DELETE CASCADE,
60     quantity integer          NOT NULL CHECK ( quantity >= 0),
61     price      money          NOT NULL CHECK ( price >= CAST(0 AS money))
62 );
63 );
64
65 CREATE TABLE IF NOT EXISTS gameStylesDist(
66     id          serial          PRIMARY KEY,
67     physicalSweeper integer      NOT NULL CHECK ( physicalSweeper
        >= 0),
68     specialSweeper integer      NOT NULL CHECK ( specialSweeper
        >= 0),
69     wall        integer          NOT NULL CHECK ( wall >= 0),
70     physicalTank integer          NOT NULL CHECK ( physicalTank >=
        0),
71     specialTank integer          NOT NULL CHECK ( specialTank >=
        0),
72     CHECK(physicalSweeper + specialSweeper + wall + physicalTank +
        specialTank = 100)
73 );
74
75 CREATE TABLE IF NOT EXISTS trainers(
76     id          serial          PRIMARY KEY,
77     nickname    varchar(32)      NOT NULL UNIQUE ,
78     gender      varchar(32)      NOT NULL CHECK ( gender in ('MALE',
        'FEMALE') ),
79     level       integer          NOT NULL CHECK ( level >= 0),
80     gameStyleID integer          REFERENCES gameStylesDist(id) ON DELETE
        SET NULL
81 );
82
83 CREATE TABLE IF NOT EXISTS trainersEntities(
84     trainerID   integer          REFERENCES trainers(id) ON DELETE
        CASCADE,
85     entityID    integer          REFERENCES entities(id) ON DELETE
        CASCADE,
86     PRIMARY KEY (trainerID, entityID)
87 );
88
89 CREATE TABLE IF NOT EXISTS orderStatuses(
90     id          serial          PRIMARY KEY,
91     name        varchar(32)      NOT NULL UNIQUE
92 );
93
94 CREATE TABLE IF NOT EXISTS orders(
95     id          serial          PRIMARY KEY,
96     trainerID   integer          NOT NULL REFERENCES trainers(id) ON
        DELETE CASCADE,

```

```

97     statusID      integer          REFERENCES orderStatuses(id) ON DELETE
98         SET NULL,
99     totalPrice    money            NOT NULL CHECK ( totalPrice >= CAST(0 AS
100         money)) DEFAULT CAST(0 AS money),
101     orderDate     date             NOT NULL CHECK (orderDate <=
102         CURRENT_DATE) DEFAULT CURRENT_DATE
103 );
104
105 CREATE TABLE IF NOT EXISTS "orderItems"(
106     orderID       integer          NOT NULL REFERENCES orders(id) ON DELETE
107         CASCADE,
108     itemID        integer          NOT NULL REFERENCES inStock(id) ON DELETE
109         CASCADE,
110     quantity      integer          NOT NULL CHECK ( quantity >= 0),
111     PRIMARY KEY (orderID, itemID)
112 );

```

5 Заполнить созданные таблицы тестовыми данными.

```

1  INSERT INTO pokemonTypes values (DEFAULT, 'NORMAL'); -- 1
2  INSERT INTO pokemonTypes values (DEFAULT, 'FIRE'); -- 2
3  INSERT INTO pokemonTypes values (DEFAULT, 'WATER'); -- 3
4  INSERT INTO pokemonTypes values (DEFAULT, 'ELECTRIC'); -- 4
5  INSERT INTO pokemonTypes values (DEFAULT, 'GRASS'); -- 5
6  INSERT INTO pokemonTypes values (DEFAULT, 'ICE'); -- 6
7  INSERT INTO pokemonTypes values (DEFAULT, 'FIGHTING'); -- 7
8  INSERT INTO pokemonTypes values (DEFAULT, 'POISON'); -- 8
9  INSERT INTO pokemonTypes values (DEFAULT, 'GROUND'); -- 9
10 INSERT INTO pokemonTypes values (DEFAULT, 'FLYING'); -- 10
11 INSERT INTO pokemonTypes values (DEFAULT, 'PSYCHIC'); -- 11
12 INSERT INTO pokemonTypes values (DEFAULT, 'BUG'); -- 12
13 INSERT INTO pokemonTypes values (DEFAULT, 'ROCK'); -- 13
14 INSERT INTO pokemonTypes values (DEFAULT, 'GHOST'); -- 14
15 INSERT INTO pokemonTypes values (DEFAULT, 'DRAGON'); -- 15
16 INSERT INTO pokemonTypes values (DEFAULT, 'DARK'); -- 16
17 INSERT INTO pokemonTypes values (DEFAULT, 'STEEL'); -- 17
18 INSERT INTO pokemonTypes values (DEFAULT, 'FAIRY'); -- 18
19
20 INSERT INTO basestats values (DEFAULT, 45, 49, 49, 65, 65, 45);
21 INSERT INTO pokemonwiki values (DEFAULT, 'Bulbasaur', 1, 'Seed Pokémon',
22     0.7, 6.9, 1);
23 INSERT INTO pokemontotypes values (1,5);
24 INSERT INTO pokemontotypes values (1,8);
25 INSERT INTO entities values (DEFAULT, 'POKEMON');
26 INSERT INTO pokemon values (1, 1, 1);
27
28 INSERT INTO basestats values (DEFAULT, 60, 62, 63, 80, 80, 60);
29 INSERT INTO pokemonwiki values (DEFAULT, 'Ivysaur', 2, 'Seed Pokémon', 1.0,
30     13.0, 2);
31 INSERT INTO pokemontotypes values (2,5);

```

```

30 INSERT INTO pokemontotypes values (2,8);
31 INSERT INTO entities values (DEFAULT, 'POKEMON');
32 INSERT INTO pokemon values (2, 2, 16);
33
34 INSERT INTO basestats values (DEFAULT, 80, 82, 83, 100, 100, 80);
35 INSERT INTO pokemonwiki values (DEFAULT, 'Venusaur', 3, 'Seed Pokémon',
    2.0, 100.0, 3);
36 INSERT INTO pokemontotypes values (3,5);
37 INSERT INTO pokemontotypes values (3,8);
38 INSERT INTO entities values (DEFAULT, 'POKEMON');
39 INSERT INTO pokemon values (3, 3, 32);
40
41 INSERT INTO basestats values (DEFAULT, 39, 52, 43, 60, 50, 65);
42 INSERT INTO pokemonwiki values (DEFAULT, 'Charmander', 4, 'Lizard Pokémon',
    0.6, 8.5, 4);
43 INSERT INTO pokemontotypes values (4,2);
44 INSERT INTO entities values (DEFAULT, 'POKEMON');
45 INSERT INTO pokemon values (4, 4, 1);
46
47 INSERT INTO basestats values (DEFAULT, 58, 64, 58, 80, 65, 80);
48 INSERT INTO pokemonwiki values (DEFAULT, 'Charmeleon', 5, 'Flame Pokémon',
    1.1, 19.0, 5);
49 INSERT INTO pokemontotypes values (5,2);
50 INSERT INTO entities values (DEFAULT, 'POKEMON');
51 INSERT INTO pokemon values (5, 5, 16);
52
53 INSERT INTO basestats values (DEFAULT, 78, 84, 78, 109, 85, 100);
54 INSERT INTO pokemonwiki values (DEFAULT, 'Charizard', 6, 'Flame Pokémon',
    1.7, 90.5, 6);
55 INSERT INTO pokemontotypes values (6,2);
56 INSERT INTO pokemontotypes values (6,10);
57 INSERT INTO entities values (DEFAULT, 'POKEMON');
58 INSERT INTO pokemon values (6, 6, 36);
59
60 INSERT INTO basestats values (DEFAULT, 44, 48, 65, 50, 64, 43);
61 INSERT INTO pokemonwiki values (DEFAULT, 'Squirtle', 7, 'Tiny Turtle
    Pokémon', 0.5, 9.0, 7);
62 INSERT INTO pokemontotypes values (7,3);
63 INSERT INTO entities values (DEFAULT, 'POKEMON');
64 INSERT INTO pokemon values (7, 7, 1);
65
66 INSERT INTO basestats values (DEFAULT, 59, 63, 80, 65, 80, 58);
67 INSERT INTO pokemonwiki values (DEFAULT, 'Wartortle', 8, 'Turtle Pokémon',
    1.0, 22.5, 8);
68 INSERT INTO pokemontotypes values (8,3);
69 INSERT INTO entities values (DEFAULT, 'POKEMON');
70 INSERT INTO pokemon values (8, 8, 16);
71
72 INSERT INTO basestats values (DEFAULT, 79, 83, 100, 85, 105, 78);
73 INSERT INTO pokemonwiki values (DEFAULT, 'Blastoise', 9, 'Shellfish
    Pokémon', 1.6, 85.5, 9);

```



```

74 INSERT INTO pokemontotypes values (9,3);
75 INSERT INTO entities values (DEFAULT, 'POKEMON');
76 INSERT INTO pokemon values (9, 9, 36);
77
78 INSERT INTO basestats values (DEFAULT, 45, 30, 35, 20, 20, 45);
79 INSERT INTO pokemonwiki values (DEFAULT, 'Caterpie', 10, 'Worm Pokémon',
    0.3, 2.9, 10);
80 INSERT INTO pokemontotypes values (10,12);
81 INSERT INTO entities values (DEFAULT, 'POKEMON');
82 INSERT INTO pokemon values (10, 10, 1);
83
84 INSERT INTO basestats values (DEFAULT, 50, 20, 55, 25, 25, 30);
85 INSERT INTO pokemonwiki values (DEFAULT, 'Metapod', 11, 'Cocoon Pokémon',
    0.7, 9.9, 11);
86 INSERT INTO pokemontotypes values (11,12);
87 INSERT INTO entities values (DEFAULT, 'POKEMON');
88 INSERT INTO pokemon values (11, 11, 7);
89
90 INSERT INTO basestats values (DEFAULT, 60, 45, 50, 90, 80, 70);
91 INSERT INTO pokemonwiki values (DEFAULT, 'Butterfree', 12, 'Butterfly
    Pokémon', 1.1, 32.0, 12);
92 INSERT INTO pokemontotypes values (12,12);
93 INSERT INTO pokemontotypes values (12,10);
94 INSERT INTO entities values (DEFAULT, 'POKEMON');
95 INSERT INTO pokemon values (12, 12, 10);
96
97 INSERT INTO basestats values (DEFAULT, 40, 35, 30, 20, 20, 50);
98 INSERT INTO pokemonwiki values (DEFAULT, 'Weedle', 13, 'Hairy Bug Pokémon',
    0.3, 3.2, 13);
99 INSERT INTO pokemontotypes values (13,12);
100 INSERT INTO pokemontotypes values (13,8);
101 INSERT INTO entities values (DEFAULT, 'POKEMON');
102 INSERT INTO pokemon values (13, 13, 1);
103
104 INSERT INTO basestats values (DEFAULT, 45, 25, 50, 25, 25, 35);
105 INSERT INTO pokemonwiki values (DEFAULT, 'Kakuna', 14, 'Cocoon Pokémon',
    0.6, 10.0, 14);
106 INSERT INTO pokemontotypes values (14,12);
107 INSERT INTO pokemontotypes values (14,8);
108 INSERT INTO entities values (DEFAULT, 'POKEMON');
109 INSERT INTO pokemon values (14, 14, 7);
110
111 INSERT INTO basestats values (DEFAULT, 65, 90, 40, 45, 80, 75);
112 INSERT INTO pokemonwiki values (DEFAULT, 'Beedrill', 15, 'Poison Bee
    Pokémon', 1.0, 29.5, 15);
113 INSERT INTO pokemontotypes values (15,12);
114 INSERT INTO pokemontotypes values (15,8);
115 INSERT INTO entities values (DEFAULT, 'POKEMON');
116 INSERT INTO pokemon values (15, 15, 10);
117
118 INSERT INTO basestats values (DEFAULT, 40, 45, 40, 35, 35, 56);

```

```

119 INSERT INTO pokemonwiki values (DEFAULT,'Pidgey', 16, 'Tiny Bird Pokémon',
    0.3, 1.8, 16);
120 INSERT INTO pokemontotypes values (16,10);
121 INSERT INTO pokemontotypes values (16,1);
122 INSERT INTO entities values (DEFAULT, 'POKEMON');
123 INSERT INTO pokemon values (16, 16, 1);
124
125 INSERT INTO basestats values (DEFAULT, 63, 60, 55, 50, 50, 71);
126 INSERT INTO pokemonwiki values (DEFAULT,'Pidgeotto', 17, 'Bird Pokémon',
    1.1, 30.0, 17);
127 INSERT INTO pokemontotypes values (17,10);
128 INSERT INTO pokemontotypes values (17,1);
129 INSERT INTO entities values (DEFAULT, 'POKEMON');
130 INSERT INTO pokemon values (17, 17, 18);
131
132 INSERT INTO basestats values (DEFAULT, 83, 80, 75, 70, 70, 101);
133 INSERT INTO pokemonwiki values (DEFAULT,'Pidgeot', 18, 'Bird Pokémon',
    1.5, 39.5, 18);
134 INSERT INTO pokemontotypes values (18,10);
135 INSERT INTO pokemontotypes values (18,1);
136 INSERT INTO entities values (DEFAULT, 'POKEMON');
137 INSERT INTO pokemon values (18, 18, 36);
138
139 INSERT INTO basestats values (DEFAULT, 30, 56, 35, 25, 35, 72);
140 INSERT INTO pokemonwiki values (DEFAULT,'Rattata', 19, 'Mouse Pokémon',
    0.3, 3.5, 19);
141 INSERT INTO pokemontotypes values (19,1);
142 INSERT INTO entities values (DEFAULT, 'POKEMON');
143 INSERT INTO pokemon values (19, 19, 1);
144
145 INSERT INTO basestats values (DEFAULT, 55, 81, 60, 50, 70, 97);
146 INSERT INTO pokemonwiki values (DEFAULT,'Raticate', 20, 'Mouse Pokémon',
    0.7, 18.5, 20);
147 INSERT INTO pokemontotypes values (20,1);
148 INSERT INTO entities values (DEFAULT, 'POKEMON');
149 INSERT INTO pokemon values (20, 20, 20);
150
151 INSERT INTO basestats values (DEFAULT, 40, 60, 30, 31, 31, 70);
152 INSERT INTO pokemonwiki values (DEFAULT,'Spearow', 21, 'Tiny Bird
    Pokémon', 0.3, 2.0, 21);
153 INSERT INTO pokemontotypes values (21,10);
154 INSERT INTO pokemontotypes values (21,1);
155 INSERT INTO entities values (DEFAULT, 'POKEMON');
156 INSERT INTO pokemon values (21, 21, 1);
157
158 INSERT INTO basestats values (DEFAULT, 65, 90, 65, 61, 61, 100);
159 INSERT INTO pokemonwiki values (DEFAULT,'Fearow', 22, 'Beak Pokémon', 1.2,
    38.0, 22);
160 INSERT INTO pokemontotypes values (22,10);
161 INSERT INTO pokemontotypes values (22,1);
162 INSERT INTO entities values (DEFAULT, 'POKEMON');

```

```

163 INSERT INTO pokemon values (22, 22, 20);
164
165 INSERT INTO basestats values (DEFAULT, 35, 60, 44, 40, 54, 55);
166 INSERT INTO pokemonwiki values (DEFAULT, 'Ekans', 23, 'Snake Pokémon', 2.0,
    6.9, 23);
167 INSERT INTO pokemontotypes values (23,8);
168 INSERT INTO entities values (DEFAULT, 'POKEMON');
169 INSERT INTO pokemon values (23, 23, 1);
170
171 INSERT INTO basestats values (DEFAULT, 60, 95, 69, 65, 79, 80);
172 INSERT INTO pokemonwiki values (DEFAULT, 'Arbok', 24, 'Cobra Pokémon', 3.5,
    65.0, 24);
173 INSERT INTO pokemontotypes values (24,8);
174 INSERT INTO entities values (DEFAULT, 'POKEMON');
175 INSERT INTO pokemon values (24, 24, 22);
176
177 INSERT INTO basestats values (DEFAULT, 35, 60, 44, 40, 54, 55);
178 INSERT INTO pokemonwiki values (DEFAULT, 'Pikachu', 25, 'Mouse Pokémon',
    0.4, 6.0, 25);
179 INSERT INTO pokemontotypes values (25,4);
180 INSERT INTO entities values (DEFAULT, 'POKEMON');
181 INSERT INTO pokemon values (25, 25, 1);
182
183 INSERT INTO basestats values (DEFAULT, 60, 90, 55, 90, 80, 110);
184 INSERT INTO pokemonwiki values (DEFAULT, 'Raichu', 26, 'Mouse Pokémon',
    0.8, 30.0, 26);
185 INSERT INTO pokemontotypes values (26,4);
186 INSERT INTO entities values (DEFAULT, 'POKEMON');
187 INSERT INTO pokemon values (26, 26, 20);
188
189 INSERT INTO basestats values (DEFAULT, 50, 75, 85, 20, 30, 40);
190 INSERT INTO pokemonwiki values (DEFAULT, 'Sandshrew', 27, 'Mouse Pokémon',
    0.6, 12.0, 27);
191 INSERT INTO pokemontotypes values (27,9);
192 INSERT INTO entities values (DEFAULT, 'POKEMON');
193 INSERT INTO pokemon values (27, 27, 1);
194
195 INSERT INTO basestats values (DEFAULT, 75, 100, 110, 45, 55, 65);
196 INSERT INTO pokemonwiki values (DEFAULT, 'Sandslash', 28, 'Mouse Pokémon',
    1.0, 29.5, 28);
197 INSERT INTO pokemontotypes values (28,9);
198 INSERT INTO entities values (DEFAULT, 'POKEMON');
199 INSERT INTO pokemon values (28, 28, 22);
200
201 INSERT INTO basestats values (DEFAULT, 50, 75, 90, 10, 35, 40);
202 INSERT INTO pokemonwiki values (DEFAULT, 'Alolan Sandshrew', 27, 'Mouse
    Pokémon', 0.7, 12.0, 28);
203 INSERT INTO pokemontotypes values (29,6);
204 INSERT INTO pokemontotypes values (29,17);
205 INSERT INTO entities values (DEFAULT, 'POKEMON');
206 INSERT INTO pokemon values (29, 29, 1);

```

```

207
208 INSERT INTO basestats values (DEFAULT, 75, 100, 120, 25, 65, 65);
209 INSERT INTO pokemonwiki values (DEFAULT, 'Alolan Sandslash', 28, 'Mouse
    Pokémon', 1.2, 55.0, 28);
210 INSERT INTO pokemontotypes values (30,6);
211 INSERT INTO pokemontotypes values (30,17);
212 INSERT INTO entities values (DEFAULT, 'POKEMON');
213 INSERT INTO pokemon values (30, 30, 22);
214
215
216 INSERT INTO entities values (DEFAULT, 'STONE');
217 INSERT INTO stonewiki values (31, 'Thunder Stone');
218 INSERT INTO evolutionchains values (25, 26, 31);
219
220 INSERT INTO entities values (DEFAULT, 'STONE');
221 INSERT INTO stonewiki values (32, 'Ice Stone');
222 INSERT INTO evolutionchains values (29, 30, 32);
223
224 INSERT INTO instock values (DEFAULT, 1, 2, 100);
225 DO $$
226 BEGIN
227     FOR i IN 2..30 LOOP
228         INSERT INTO instock values (DEFAULT, i, floor(random() *
            101)::integer, (floor(random() * 101)::numeric + 100)::money);
229     END LOOP;
230 END$$;
231
232 INSERT INTO gamestylesdist values (DEFAULT, 23, 63, 14, 0, 0);
233 INSERT INTO trainers values (DEFAULT, 'Ash', 'MALE', 1, 1);
234 INSERT INTO trainersentities values (1, 1);
235 INSERT INTO trainersentities values (1, 2);
236
237 INSERT INTO gamestylesdist values (DEFAULT, 20, 63, 17, 0, 0);
238 INSERT INTO trainers values (DEFAULT, 'Aqua', 'MALE', 2, 2);
239 INSERT INTO trainersentities values (2, 3);
240 INSERT INTO trainersentities values (2, 4);
241
242 INSERT INTO orderStatuses values (DEFAULT, 'PROCESSING');
243 INSERT INTO orderStatuses values (DEFAULT, 'CANCELLED');
244 INSERT INTO orderStatuses values (DEFAULT, 'COMPLETED');
245
246 DO $$
247 BEGIN
248     FOR i IN 1..30 LOOP
249         INSERT INTO orders values (DEFAULT, floor(random() * 2 +
            1)::integer, floor(random() * 3 + 1)::integer, DEFAULT,
            DEFAULT);
250     END LOOP;
251 END$$;
252
253 DO $$

```

```

254 BEGIN
255     FOR i IN 1..30 LOOP
256         FOR j IN 1..floor(random() * 30 + 1)::integer LOOP
257             INSERT INTO "orderItems" values (i, j, floor(random() * 5
                + 1)::integer);
258         END LOOP;
259     END LOOP;
260 END$$;
261
262 INSERT INTO "orderItems" values (22, 28, 2);

```

6 Сделать скрипты для:

6.1 создания/удаления объектов базы данных;

```

1 DROP TABLE IF EXISTS trainersEntities;
2 DROP TABLE IF EXISTS orderItems;
3 DROP TABLE IF EXISTS evolutionChains;
4 DROP TABLE IF EXISTS pokemonToTypes;
5 DROP TABLE IF EXISTS pokemon;
6 DROP TABLE IF EXISTS stoneWiki;
7 DROP TABLE IF EXISTS pokemonWiki;
8 DROP TABLE IF EXISTS baseStats;
9 DROP TABLE IF EXISTS pokemonTypes;
10 DROP TABLE IF EXISTS inStock CASCADE ;
11 DROP TABLE IF EXISTS orderStatuses CASCADE ;
12 DROP TABLE IF EXISTS orders CASCADE ;
13 DROP TABLE IF EXISTS entities CASCADE ;
14 DROP TABLE IF EXISTS wikiTypes CASCADE ;
15 DROP TABLE IF EXISTS trainers CASCADE ;
16 DROP TABLE IF EXISTS gameStylesDist CASCADE ;

```

6.2 заполнения/удаления созданных таблиц;

```

1 TRUNCATE TABLE pokemontypes RESTART IDENTITY CASCADE;
2 TRUNCATE TABLE pokemontotypes RESTART IDENTITY CASCADE;
3 TRUNCATE TABLE pokemonwiki RESTART IDENTITY CASCADE;
4 TRUNCATE TABLE basestats RESTART IDENTITY CASCADE;
5 TRUNCATE TABLE pokemon RESTART IDENTITY CASCADE;
6 TRUNCATE TABLE entities RESTART IDENTITY CASCADE;
7 TRUNCATE TABLE stonewiki RESTART IDENTITY CASCADE;
8 TRUNCATE TABLE evolutionchains RESTART IDENTITY CASCADE;
9 TRUNCATE TABLE "orderItems" RESTART IDENTITY CASCADE;
10 TRUNCATE TABLE instock RESTART IDENTITY CASCADE;
11 TRUNCATE TABLE trainersentities RESTART IDENTITY CASCADE;
12 TRUNCATE TABLE trainers RESTART IDENTITY CASCADE;
13 TRUNCATE TABLE gamestylesdist RESTART IDENTITY CASCADE;
14 TRUNCATE TABLE orders RESTART IDENTITY CASCADE;

```

```
15 TRUNCATE TABLE orderstatuses RESTART IDENTITY CASCADE;
```

7 Добавить в базу данных триггеры для обеспечения комплексных ограничений целостности.

```
1
2
3 CREATE OR REPLACE FUNCTION updateTotalPrice() RETURNS TRIGGER AS $$
4 BEGIN
5     UPDATE "orders" SET totalprice = (SELECT SUM(instock.price *
6         "orderItems".quantity) FROM "orderItems"
7         JOIN instock on
8             instock.entityid =
9             "orderItems".itemid
10        WHERE "orderItems".orderid
11            = NEW.orderid
12        GROUP BY
13            "orderItems".orderid)
14        WHERE "orders".id =
15            NEW.orderid;
16
17 RETURN NEW;
18 END;
19 $$
20
21 LANGUAGE plpgsql;
22
23
24 CREATE trigger totalPriceTrigger
25 after INSERT on "orderItems"
26 FOR EACH ROW EXECUTE PROCEDURE updateTotalPrice();
27
28
29 CREATE OR REPLACE FUNCTION updateTotalItemQuantity() RETURNS TRIGGER AS $$
30 BEGIN
31     IF ( (SELECT COUNT(*) FROM "orderItems" WHERE "orderItems".itemid =
32         NEW.itemid and "orderItems".orderid = NEW.orderid) >= 1 )
33     THEN
34         UPDATE "orderItems" SET quantity = quantity + NEW.quantity WHERE
35             itemid = NEW.itemid and "orderItems".orderid = NEW.orderid;
36         RETURN NULL;
37     ELSE
38         RETURN NEW;
39     END IF;
40 END;
41 $$ LANGUAGE plpgsql;
42
43
44 CREATE OR REPLACE FUNCTION resetInstoke() RETURNS TRIGGER AS $$
45 declare mviews record;
46 BEGIN
```

```

37     IF (new.statusid = 2)
38         THEN
39             for mviews in
40                 select quantity, itemid
41                 from "orderItems" where "orderItems".orderid=new.id
42                 order by 1
43             loop
44                 perform increaser(mviews.itemid, mviews.quantity);
45             end loop;
46             update "orderItems" set quantity=0 where
47                 "orderItems".orderid=new.id;
48         END IF;
49     RETURN NEW;
50 END
51 $$ LANGUAGE plpgsql;
52
53 CREATE OR REPLACE TRIGGER resetInstokeTrigger
54 BEFORE update ON orders
55 FOR EACH ROW EXECUTE PROCEDURE resetInstoke();

```

8 Реализовать функции и процедуры на основе описания бизнес-процессов (из этапа №1).

```

1 CREATE or replace FUNCTION addNewOrder(
2   trainerId int)
3 RETURNS void AS $$
4   INSERT INTO orders values (DEFAULT, trainerId,
5   1,
6   DEFAULT, DEFAULT);
7 $$ LANGUAGE SQL;
8
9 CREATE or replace FUNCTION addNewEntity(newEntityType varchar(32))
10 RETURNS void AS $$
11     insert into entities values (default, newEntityType);
12 $$ LANGUAGE SQL;
13
14 CREATE or replace FUNCTION addNewStone(name varchar(32))
15 RETURNS void AS $$
16     select addNewEntity('STONE');
17     insert into stonewiki values ((select id from entities order by
18         entities.id DESC limit 1), name);
19 $$ LANGUAGE SQL;
20
21 select* from entities;
22
23 CREATE or replace FUNCTION addNewItenToStore(
24   wareId int, count int, price money)

```

```

25 RETURNS void AS $$
26 begin
27     if((select count(*) from instock where instock.entityid=wareId)>0)
28         then
29             perform increaser(wareId, count);
30         else
31             insert into instock values (default, wareId, count, price);
32         end if;
33 end;
34 $$ LANGUAGE plpgsql;
35
36 CREATE or replace FUNCTION decreaser(
37 itemId int, itemCount int)
38 RETURNS void AS $$
39 update instock set quantity=quantity-itemCount where
40     instock.entityid=itemId;
41 $$ LANGUAGE SQL;
42
43 CREATE or replace FUNCTION increaser(
44 itemId int, itemCount int)
45 RETURNS void AS $$
46 update instock set quantity=quantity+itemCount where
47     instock.entityid=itemId;
48 $$ LANGUAGE SQL;
49
50 CREATE or replace FUNCTION addToOrder(
51 orderId int, itemId int, itemCount int)
52 RETURNS void AS $$
53 insert into "orderItems" values (orderId, itemId, itemCount);
54 select decreaser(itemId, itemCount);
55 $$ LANGUAGE SQL;
56
57 CREATE or replace FUNCTION setOrderStatus(
58 orderID int, orderStatus int)
59 RETURNS void AS $$
60 update orders set statusid=orderStatus where orders.id=orderID;
61 $$ LANGUAGE SQL;
62
63 CREATE or replace FUNCTION addTrainer(
64 name varchar(32), gender varchar(32), trainerLevel integer, styleId
65 integer)
66 RETURNS void AS $$
67 insert into trainers values (default, name, gender, trainerLevel, styleId);
68 $$ LANGUAGE SQL;
69
70 CREATE or replace FUNCTION addPokemonToTrainer(
71 pokemonId integer, trainerId integer)
72 RETURNS void AS $$
73 insert into trainersentities values (trainerId, (select pokemon.entityid
74     from pokemon where pokemon.pokemonwikiid=pokemonId));
75 $$ LANGUAGE SQL;

```



```

1  /*проверка наличия такого числа товаров*/
2  CREATE or replace FUNCTION checkLimit(
3  itemId int, itemCount int )
4  RETURNS bool AS $$
5      begin
6          return ((select instock.quantity from instock where
7                     instock.entityid=itemId) >= itemCount);
8      end;
9  $$ LANGUAGE plpgsql;
10
11 /*узнать тип*/
12 CREATE or replace FUNCTION whatTypeIs(
13 pokId int)
14 RETURNS setof pokemontotypes AS $$
15 (select * from pokemontotypes where pokemontotypes.pokemonid=pokId);
16 $$ LANGUAGE SQL;
17
18 /*узнать статус заказа*/
19 CREATE or replace FUNCTION knowOrderStatus(
20 orderId int)
21 RETURNS int AS $$
22 (select statusid from orders where orders.id=orderId);
23 $$ LANGUAGE SQL;
24
25 /*добавить харки тренера*/
26 CREATE or replace FUNCTION addTrainerStyle(
27 HisPhysicalsWeeper integer, HisSpecialSweeper integer,
28 hisWall integer, hisPhysicalTank integer, hisSpecialTank integer )
29 RETURNS integer AS $$
30     begin
31         INSERT INTO gamestylesdist values (default, HisPhysicalsWeeper,
32                                             HisSpecialSweeper, hisWall, hisPhysicalTank, hisSpecialTank);
33         return (select gamestylesdist.id from gamestylesdist order by
34                  gamestylesdist.id desc limit 1);
35     end;
36 $$ LANGUAGE plpgsql;

```

9 Произвести анализ использования созданной базы данных:

Чаще всего используются функции добавления, удаления, поиска предметов в заказе, а также сами заказы. Все остальные таблицы создаются единожды и используются для поиска.

10 Создать индексы и доказать, что они полезны для вашей базы данных:

```
1 CREATE INDEX national_number_index
2     ON pokemonwiki (nationalnum)
```

Индекс необходим для быстрого поиска по nationalNum, что является важной характеристикой покемона и тренеры захотят часто искать именно по этому полю.