

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Отчет по предмету Архитектура информационных систем
2 лабораторная работа
Приложение для автоматизированного проведения тестирования

Вычиков Д.Д

1 Диаграмма сущность-связь

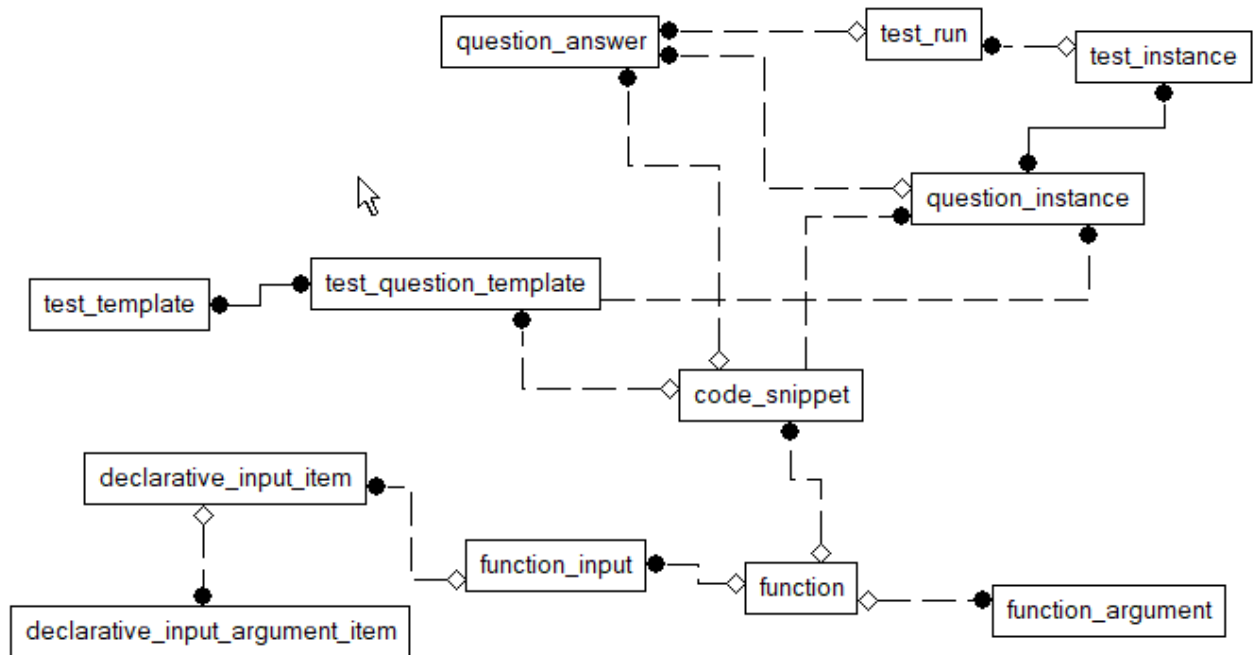


Figure 1: Диаграмма сущность-связь

На диаграмма отображены следующие сущности:

1. test_template - Шаблон теста
2. test_question_template - Шаблон вопроса
3. code_snippet - Объект для представления кода процедуры
4. test_instance - Тестовое событие
5. question_instance - Вопрос, принадлежащий тестовому событию
6. test_run - Прохождение тестирования
7. question_answer - Ответ на вопрос теста
8. function - Процедура
9. function_argument - Аргумент процедуры
10. function_input - Набор тестовых параметров процедуры
11. declarative_input_item
12. declarative_input_argument_item

Связи между сущностями

1. Тестовый шаблон может иметь несколько тестовых вопросов
2. Тестовый вопрос может принадлежать к нескольким тестам
3. Вопрос тестового события принадлежит к одному шаблону вопроса
4. Тестовое событие содержит несколько вопросов
5. Ответ на вопрос принадлежит только к одному событию
6. Функция может иметь сколько угодно аргументов, реализаций и тестирующих данных

Были определены следующие домены

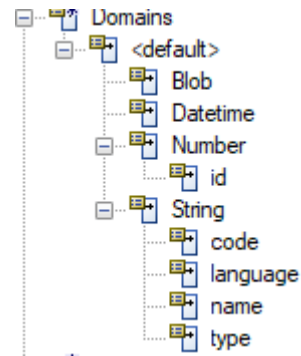


Figure 2: Определенные домены

2 Модель данных, основанная на ключах

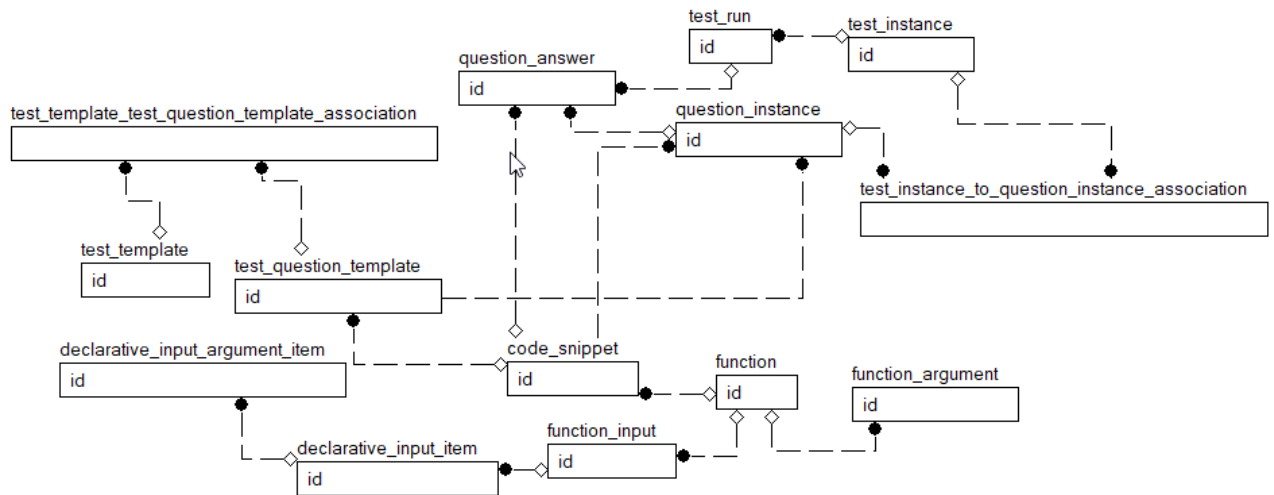


Figure 3: Модель данных, основанная на ключах

На данной схеме связи типа многие-ко-многим между парами сущностей `test_template` и `test_question_template`, `test_instance` и `question_instance` были заменены двумя связями, выраженными с помощью промежуточных таблиц `test_template_test_question_template_association` и `test_instance_to_question_instance_association`.

3 Полная атрибутивная модель

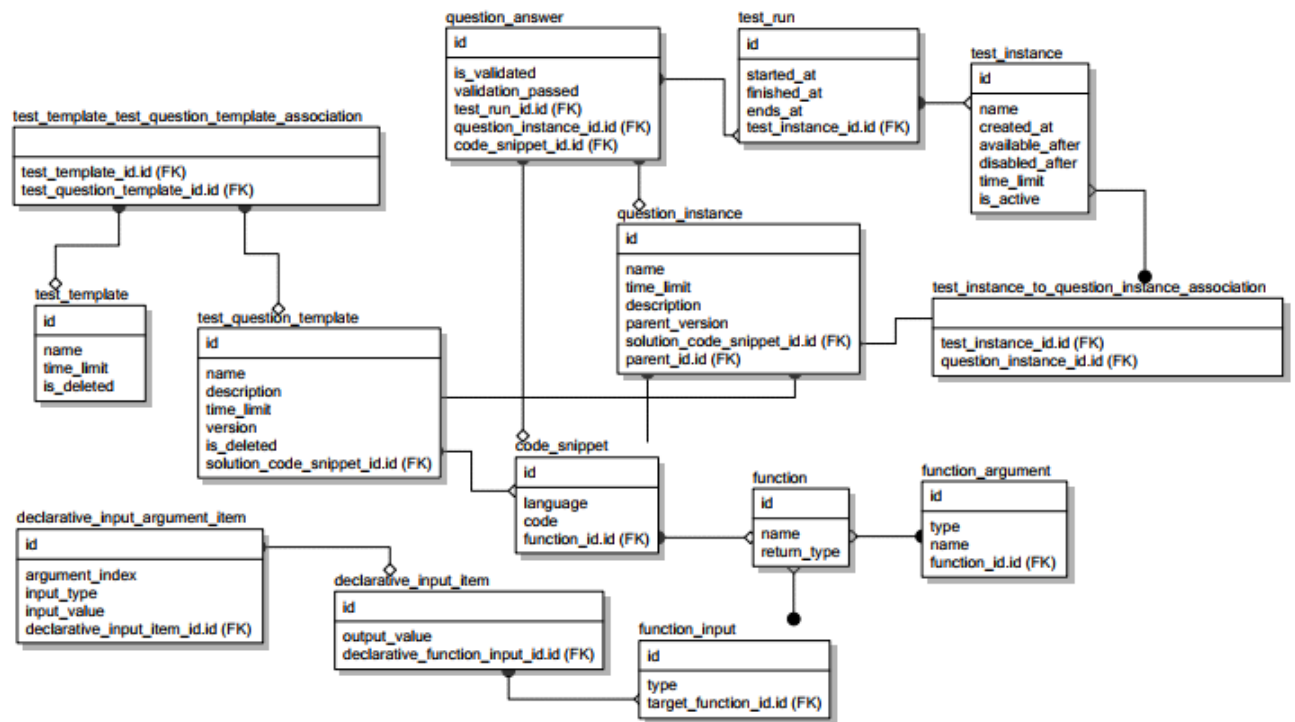


Figure 4: Полная атрибутивная модель

На данной модели были добавлены атрибуты к сущностям.

Атрибуты сущности test_template

1. name - Название
2. time_limit - Ограничение по времени
3. is_deleted - Маркер удаления

Атрибуты test_question_template

1. name - Название
2. time_limit - Ограничение по времени
3. is_deleted - Маркер удаления
4. description - Описание вопроса
5. version - Версия вопроса
6. solution_code_snippet_id - Первичный ключа сущности solution_code_snippet

Атрибуты сущности test_template_test_question_template_association

1. test_template_id - Первичный ключ сущности test_template
2. test_question_template_id - Первичный ключ сущности test_question_template

Атрибуты сущности code_snippet

1. language - Язык программирования
2. code - Код решения
3. function_id - Первичный ключ сущности function

Аттрибуты сущности function

1. name - Название
2. return_type - Тип возвращаемого значения

Аттрибуты сущности function_argument

1. type - Типа аргумента
2. name - Название
3. function_id - Первичный ключ сущности function

Аттрибуты сущности function_input

1. type - Дискриминатор объекта (для наследования)
2. target_function_id - Первичный ключ сущности function

Аттрибуты сущности declarative_input_item

1. output_value - Ожидаемое возвращаемое значение
2. declarative_function_input_id - Первичный ключ сущности function_input

Аттрибуты сущности declarative_input_argument_item

1. argument_index - Индекс аргумента
2. input_type - Тип входного параметра
3. input_value Входное значение
4. declarative_input_item_id - Первичный ключ сущности declarative_input_item

Аттрибуты сущности question_instance

1. name - Название
2. time_limit - Ограничение по времени
3. description - Описание вопроса
4. parent_version - Версия вопроса
5. parent_id - Первичный ключ сущности test_question_template
6. solution_code_snippet_id - Первичный ключа сущности solution_code_snippet

Аттрибуты сущности test_instance

1. name - Название
2. time_limit - Ограничение по времени
3. is_active - Маркер доступности
4. created_at - Дата и время создания
5. available_after - Дата, после которой событие станет доступным для прохождения
6. disabled_after - Дата, после которой событие будет завершено

Аттрибуты сущности test_run

1. started_at - Дата и время начала прохождения теста
2. finished_at - Дата и время фактического завершения теста
3. ends_at - Дата и время ожидаемого завершения теста
4. test_instance_id - Первичный ключ сущности test_instance

Атрибуты сущности question_answer

1. is_validated - Был ли ответ проверен
2. validation_passed - Прошел ли ответ проверку
3. test_run_id - Первичный ключ сущности test_run
4. question_instance_id - Первичный ключ сущности question_instance
5. code_snippet_id - Первичный ключ сущности code_snippet

4 Трансформационная модель

В данной модели было добавлено представление AnswerView для отображения результатов прохождения тестирования

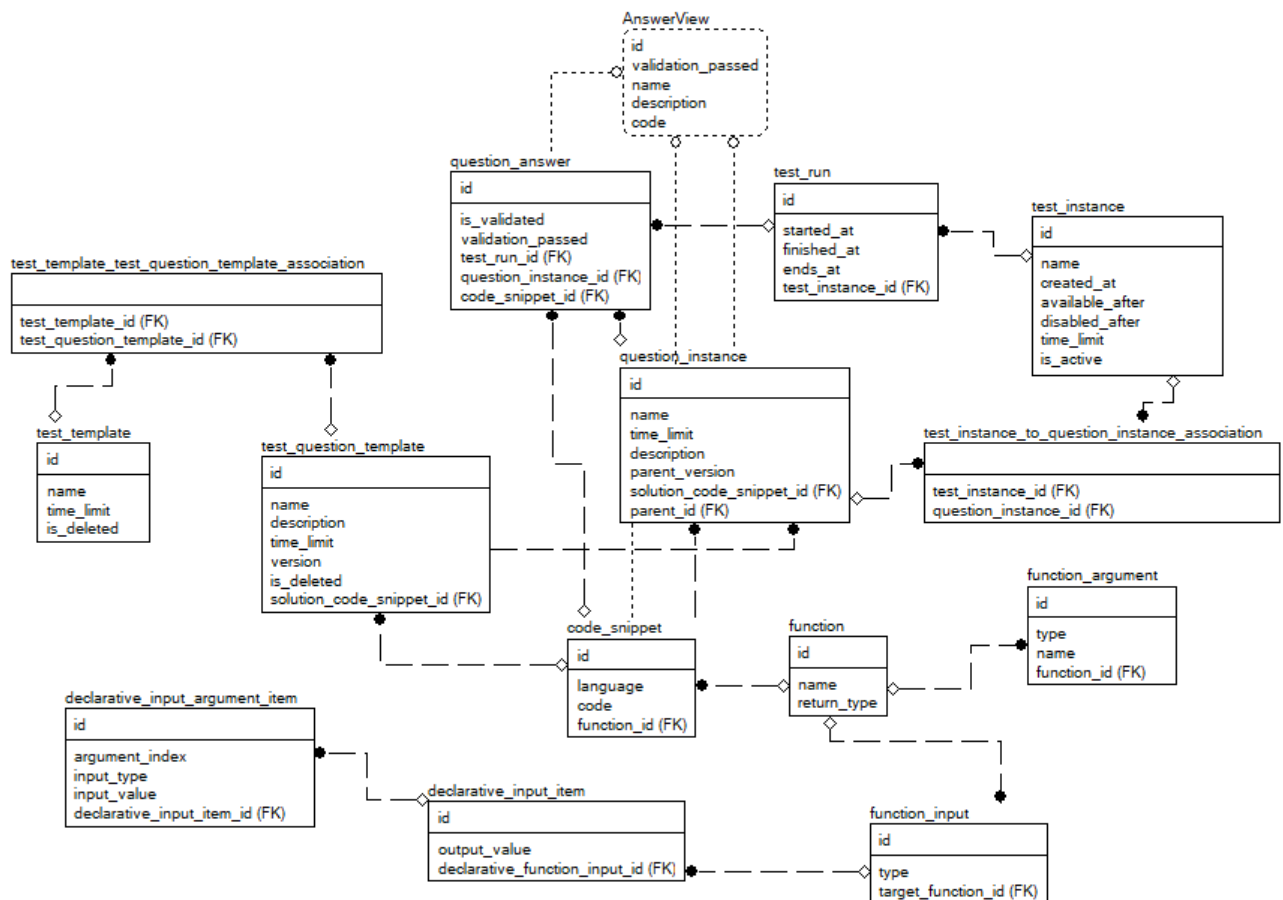


Figure 5: Трансформационная модель

5 Модель СУБД

В качестве СУБД была использована MySQL.

Для генерации созданной модели даны был получен следующий скрипт:

```
CREATE TABLE code_snippet
(
    id integer NOT NULL,
    language INTEGER NULL,
    code text(65535) NULL,
    function_id integer NULL
```

```

)
;

ALTER TABLE code_snippet
    ADD PRIMARY KEY (id)
;

CREATE TABLE declarative_input_argument_item
(
    id integer NOT NULL,
    argument_index integer NULL,
    input_type VARCHAR(20) NULL,
    input_value text(65535) NULL,
    declarative_input_item_id integer NULL
)
;

ALTER TABLE declarative_input_argument_item
    ADD PRIMARY KEY (id)
;

CREATE TABLE declarative_input_item
(
    id integer NOT NULL,
    output_value text(65535) NULL,
    declarative_function_input_id integer NULL
)
;

ALTER TABLE declarative_input_item
    ADD PRIMARY KEY (id)
;

CREATE TABLE function
(
    id integer NOT NULL,
    name varchar(100) NULL,
    return_type char(13) NULL
)
;

ALTER TABLE function
    ADD PRIMARY KEY (id)
;

CREATE TABLE function_argument
(
    id integer NOT NULL,
    type char(13) NULL,
    name varchar(100) NULL,
    function_id integer NULL
)
;

ALTER TABLE function_argument
    ADD PRIMARY KEY (id)
;

CREATE TABLE function_input
(
    id integer NOT NULL,

```

```

        type varchar(50) NULL,
        target_function_id integer NULL
    )
;

ALTER TABLE function_input
    ADD PRIMARY KEY (id)
;

CREATE TABLE question_answer
(
    id integer NOT NULL,
    is_validated tinyint NULL,
    validation_passed tinyint NULL,
    test_run_id integer NULL,
    question_instance_id integer NULL,
    code_snippet_id integer NULL
)
;

ALTER TABLE question_answer
    ADD PRIMARY KEY (id)
;

CREATE TABLE question_instance
(
    id integer NOT NULL,
    name varchar(100) NULL,
    time_limit integer NULL,
    description text(65535) NULL,
    parent_version bigint NOT NULL,
    solution_code_snippet_id integer NOT NULL,
    parent_id integer NOT NULL
)
;

ALTER TABLE question_instance
    ADD PRIMARY KEY (id)
;

CREATE TABLE test_instance
(
    id integer NOT NULL,
    name varchar(100) NULL,
    created_at datetime NULL,
    available_after datetime NULL,
    disabled_after datetime NULL,
    time_limit integer NULL,
    is_active tinyint NULL
)
;

ALTER TABLE test_instance
    ADD PRIMARY KEY (id)
;

CREATE TABLE test_instance_to_question_instance_association
(
    test_instance_id integer NULL,
    question_instance_id integer NULL
)

```



```

;

CREATE TABLE test_question_template
(
    id integer NOT NULL,
    name varchar(100) NULL,
    description text(65535) NULL,
    time_limit integer NULL,
    version bigint NULL,
    is_deleted tinyint NULL,
    solution_code_snippet_id integer NULL
)
;

ALTER TABLE test_question_template
    ADD PRIMARY KEY (id)
;

CREATE TABLE test_run
(
    id integer NOT NULL,
    started_at datetime NULL,
    finished_at datetime NULL,
    ends_at datetime NULL,
    test_instance_id integer NULL
)
;

ALTER TABLE test_run
    ADD PRIMARY KEY (id)
;

CREATE TABLE test_template
(
    id integer NOT NULL,
    name varchar(100) NULL,
    time_limit integer NULL,
    is_deleted tinyint NULL
)
;

ALTER TABLE test_template
    ADD PRIMARY KEY (id)
;

CREATE TABLE test_template_test_question_template_association
(
    test_template_id integer NULL,
    test_question_template_id integer NULL
)
;

CREATE VIEW AnswerView ( id , validation_passed , name , description , code )
AS
    SELECT question_answer.id , question_answer.validation_passed ,
        question_instance.name , question_instance.description ,
        code_snippet.code
        FROM question_answer , question_instance , code_snippet
;

ALTER TABLE code_snippet

```

```

        ADD FOREIGN KEY code_snippet_ibfk_1 (function_id) REFERENCES
            function(id)
    ;

ALTER TABLE declarative_input_argument_item
    ADD FOREIGN KEY declarative_input_argument_item_ibfk_1 (
        declarative_input_item_id) REFERENCES declarative_input_item(
            id)
    ;

ALTER TABLE declarative_input_item
    ADD FOREIGN KEY declarative_input_item_ibfk_1 (
        declarative_function_input_id) REFERENCES function_input(id)
    ;

ALTER TABLE function_argument
    ADD FOREIGN KEY function_argument_ibfk_1 (function_id)
        REFERENCES function(id)
    ;

ALTER TABLE function_input
    ADD FOREIGN KEY function_input_ibfk_1 (target_function_id)
        REFERENCES function(id)
    ;

ALTER TABLE question_answer
    ADD FOREIGN KEY question_answer_ibfk_3 (code_snippet_id)
        REFERENCES code_snippet(id)
    ;

ALTER TABLE question_answer
    ADD FOREIGN KEY question_answer_ibfk_2 (question_instance_id)
        REFERENCES question_instance(id)
    ;

ALTER TABLE question_answer
    ADD FOREIGN KEY question_answer_ibfk_1 (test_run_id) REFERENCES
        test_run(id)
    ;

ALTER TABLE question_instance
    ADD FOREIGN KEY question_instance_ibfk_2 (parent_id) REFERENCES
        test_question_template(id)
    ;

ALTER TABLE question_instance
    ADD FOREIGN KEY question_instance_ibfk_1 (
        solution_code_snippet_id) REFERENCES code_snippet(id)
    ;

ALTER TABLE test_instance_to_question_instance_association
    ADD FOREIGN KEY
        test_instance_to_question_instance_association_ibfk_2 (
            question_instance_id) REFERENCES question_instance(id)
    ;

ALTER TABLE test_instance_to_question_instance_association
    ADD FOREIGN KEY
        test_instance_to_question_instance_association_ibfk_1 (
            test_instance_id) REFERENCES test_instance(id)
    ;

```

```

ALTER TABLE test_question_template
    ADD FOREIGN KEY test_question_template_ibfk_1 (
        solution_code_snippet_id) REFERENCES code_snippet(id)
;

ALTER TABLE test_run
    ADD FOREIGN KEY test_run_ibfk_1 (test_instance_id) REFERENCES
        test_instance(id)
;

ALTER TABLE test_template_test_question_template_association
    ADD FOREIGN KEY
        test_template_test_question_template_association_ibfk_2 (
            test_question_template_id) REFERENCES test_question_template(
                id)
;

ALTER TABLE test_template_test_question_template_association
    ADD FOREIGN KEY
        test_template_test_question_template_association_ibfk_1 (
            test_template_id) REFERENCES test_template(id)
;

```

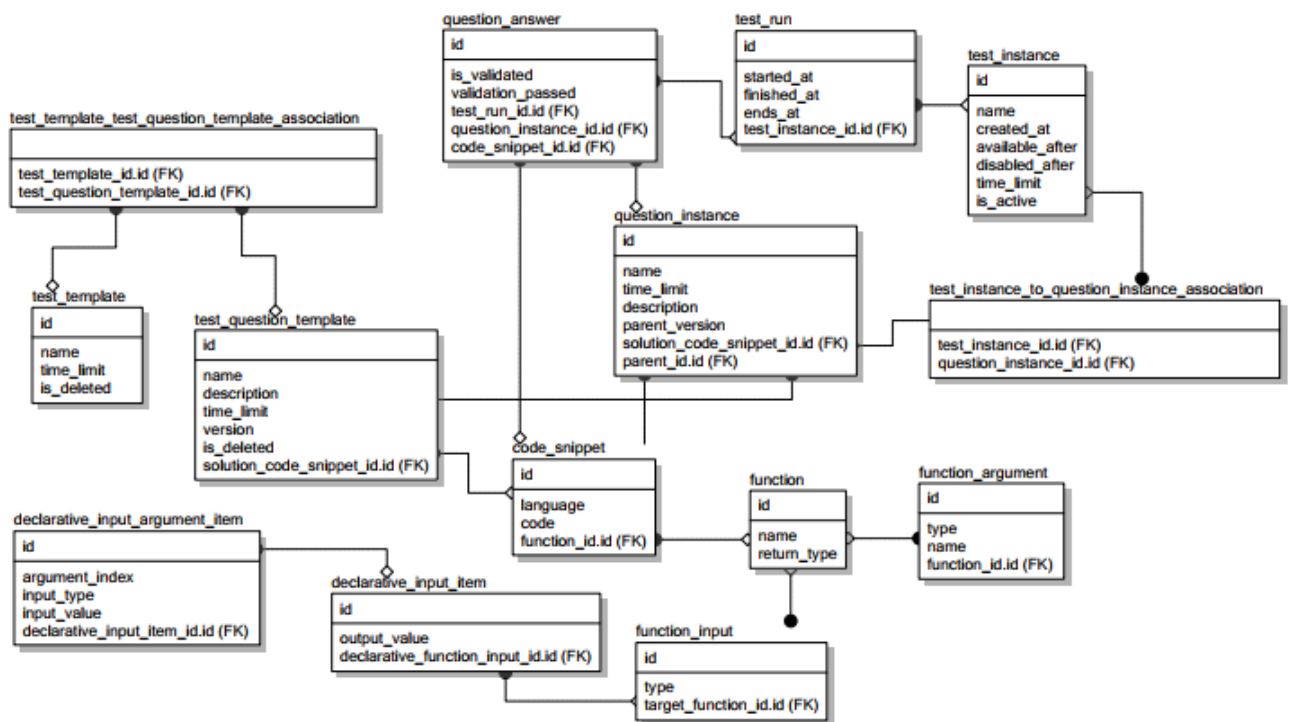


Figure 6: Модель, полученная с помощью обратной генерации

6 Автодокументация

В данном разделе будут представлены отчеты, автоматически сгенерированные с помощью ERWin.

Table Reports-Table-Comment

Table	
Name	Comment
code snippet	Код процедуры
declarative input argument item	Заданное значение аргумента функции
declarative input item	Заданный набор тестовых данных
function	Процедура
function argument	Аргумент процедуры
function input	Тестируемые данные процедуры
question answer	Ответ на вопрос
question instance	Вопрос тестового события
test instance	Тестовое событие
test_instance_to_question_instance_association	Промежуточная таблица между событиями и вопросами событий
test question template	Шаблон тестового вопроса
test run	Прохождение теста
test template	Шаблон теста
test_template_test_question_template_association	Промежуточная таблица между шаблонами тестов и шаблонами вопросов

Figure 7: Отчет о созданных таблицах базы данных

Column Reports-Column-Datatype-Null Option

Column		
Name	Datatype	Null Option
id	integer	NOT NULL
language	INTEGER	NULL
code	text(65535)	NULL
function_id	integer	NULL
id	integer	NOT NULL
argument index	integer	NULL
input type	VARCHAR(20)	NULL
input value	text(65535)	NULL
declarative input item id	integer	NULL
id	integer	NOT NULL
output value	text(65535)	NULL
declarative function input id	integer	NULL
id	integer	NOT NULL
name	varchar(100)	NULL
return type	char(13)	NULL
id	integer	NOT NULL
type	char(13)	NULL
name	varchar(100)	NULL
function_id	integer	NULL
id	integer	NOT NULL
type	varchar(50)	NULL
target function id	integer	NULL
id	integer	NOT NULL
is validated	tinyint	NULL
validation passed	tinyint	NULL
test run id	integer	NULL
question instance id	integer	NULL
code snippet id	integer	NULL
id	integer	NOT NULL
name	varchar(100)	NULL
time limit	integer	NULL
description	text(65535)	NULL
parent version	bigint	NOT NULL

Figure 8: Отчет о созданных столбцах базы данных, часть 1

solution code snippet id	integer	NOT NULL
parent id	integer	NOT NULL
id	integer	NOT NULL
name	varchar(100)	NULL
created at	datetime	NULL
available after	datetime	NULL
disabled after	datetime	NULL
time limit	integer	NULL
is active	tinyint	NULL
id	integer	NOT NULL
name	varchar(100)	NULL
description	text(65535)	NULL
time limit	integer	NULL
version	bigint	NULL
is deleted	tinyint	NULL
solution code snippet id	integer	NULL
id	integer	NOT NULL
started at	datetime	NULL
finished at	datetime	NULL
ends at	datetime	NULL
test instance id	integer	NULL
id	integer	NOT NULL
name	varchar(100)	NULL
time limit	integer	NULL
is deleted	tinyint	NULL
test template id	integer	NULL
test question template id	integer	NULL
test instance id	integer	NULL
question instance id	integer	NULL

Figure 9: Отчет о созданных столбцах базы данных, часть 2

Domain Reports-Physical Domain-Default-Validation-Column

Domain
Name
default
Blob
Datetime
Number
id
String
code
language
name
type

Figure 10: Отчет об определенных доменах

Used by Column(s) of "Datetime" Domain
Name
available after
created at
disabled after
ends at
finished at
started at

Figure 11: Колонки, определенные на домене DateTime

Used by Column(s) of "Number" Domain	
Name	
id	
language	
argument index	
id	
id	
is_validated	
validation passed	
parent version	
time limit	
is_active	
time limit	
is_deleted	
time limit	
version	

Figure 12: Колонки, определенные на домене Number