Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт дистанционного образования

УТВЕРЖДЕНО:
Директор ИДО - проректор по РДО
М.О. Шепель

Оценочные материалы по дисциплине

Основы SQL

по направлению подготовки

09.04.03 Прикладная информатика

Направленность (профиль) подготовки:
«Науки о данных»

Форма обучения
Очная

Квалификация
Магистр

Год приема
2024

СОГЛАСОВАНО:
Руководитель ОПОП
Д.Д. Даммер

Председатель УМК
С.Б. Велединская

Томск – 2024

1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-1 - способность самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте;

ОПК-3 - способность анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями;

ОПК-6 - способность исследовать современные проблемы и методы прикладной информатики и развития информационного общества;

ПК-1 - способность управлять получением, хранением, передачей, обработкой больших данных

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-1.2 Знает основы определения взаимосвязей, закономерностей, обобщения и абстрагирования фундаментальных моделей, законов и методик для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте;

ИОПК-3.2 Знает основы работы с различными видами информации с помощью различных средств информационных и коммуникационных технологий;

ИОПК-6.1 Знает основы методов анализа прикладной области, информационных потребностей, технологий сбора, накопления, обработки, передачи и распространения информации;

ИПК-1.1 Знает основы интеграции больших данных с системами хранения данных организации.

2. Оценочные материалы текущего контроля и критерии оценивания

№	**Этапы формирования компетенций (разделы дисциплины/модуля/практики)**	**Код и наименование результатов обучения**	**Вид оценочного средства** (тесты, задания, кейсы, вопросы и др.)
	Тема 1. Как выгружать данные из файлов разных форматов Работа с текстовыми файлами. Работа с файлами Excel. JSON. Что это? JSON. Открываем JSON-файл и извлекаем данные. SON. Работаем с pandas. Из JSON в pandas. JSON. Работаем с pandas. Из pandas в JSON. XML. Что это? XML. Контент XML-файла. XML. Загружаем, создаем, сохраняем	ИОПК-1.2; ИОПК-3.2	Тест Контрольная работа Домашнее задание Проект
	Тема 2. Как получать данные из веб-источников и API Веб-запросы. Библиотека requests. Парсинг сайтов. Библиотека BeautifulSoup. Работа с API. Как настроить регулярную выгрузку данных. Встроенная функция map(). Встроенная функция filter()	ИОПК-3.2; ИПК-1.1	Тест Контрольная работа Домашнее задание Проект

	Тема 3. Аттестация 1. Python Теоретическая часть. Практические задания с проверкой кода. Практические задания в Jupyter Notebook.	ИПК-1.1	Тест Контрольная работа Домашнее задание Проект
	Тема 4. Привет, SQL! Что такое базы данных? Что такое SQL? Что такое Metabase?	ИОПК-3.2;	Тест Контрольная работа Домашнее задание Проект
	Тема 5. Основы SQL Получаем все данные из таблицы. Фильтруем строки.Сортировка. Ограничение вывода.	ИОПК-6.1	Тест Контрольная работа Домашнее задание Проект
	Тема 6. Агрегатные функции Знакомимся с данными. Убираем повторяющиеся значения. Агрегатные функции. Группировка. Фильтрация агрегированных строк.	ИОПК-1.2;	Тест Контрольная работа Домашнее задание Проект
	Тема 7. Соединение таблиц Знакомимся с данными. Соединение таблиц по ключу. Знакомимся с JOIN. Фильтрация и агрегатные функции. Способы соединения таблиц	ИОПК-1.2;	Тест Контрольная работа Домашнее задание Проект
	Тема 8. Сложные объединения Знакомимся с данными. UNION. UNION и ограничение типов данных. UNION ALL и промежуточные итоги. UNION и дополнительные условия. UNION и ручная генерация. EXCEPT. INTERSECT	ИОПК-1.2	Тест Контрольная работа Домашнее задание Проект
	Тема 9. Типы данных Типы данных в PostgreSQL. Даты: основные типы. Функции и операторы для работы с датами. Строковые данные: основные типы. Функции и операторы для работы со строками.	ИОПК-6.1;	Тест Контрольная работа Домашнее задание Проект
	Тема 10. Анализ вакансий из HeadHunter Работа с базой данных из Python. Введение. Знакомство с данными. Предварительный анализ данных. Детальный анализ вакансий. Анализ работодателей. Предметный анализ	ИОПК-3.2; ИПК-1.1	Тест Контрольная работа Домашнее задание Проект
	Тема 11. Аттестация 2. SQL Аттестационное тестирование	ИПК-1.1	Тест

Примеры элементов текущего контроля:

Задание 1 (ИОПК-3.2)

Выберите верное утверждение об использовании библиотеки pandas для работы с JSON-файлами:

1. pandas не поддерживает работу с JSON-файлами
2. pandas использует метод read_json() для чтения JSON-файлов
3. pandas использует метод read_csv() для чтения JSON-файлов
4. pandas использует метод to_json() только для чтения JSON-файлов

Задание 2 (ИОПК-6.1)

Выберите верное утверждение о формате XML:

1. XML используется исключительно для хранения текстовых данных
2. XML поддерживает вложенные структуры данных
3. XML не поддерживает атрибуты для элементов
4. XML не может быть использован для передачи данных

Задание 3 (ИОПК-1.2)

Выберите верное утверждение об использовании библиотеки requests:

1. requests не поддерживает работу с API
2. requests поддерживает работу только с текстовыми данными
3. requests позволяет отправлять HTTP-запросы и получать ответы
4. requests не поддерживает работу с веб-сайтами

Задание 4 (ИПК-1.1)

Какой тип данных используется для хранения строковых значений в pandas?

1. int
2. float
3. str
4. bool

Домашнее задание

Задание 1 (ИОПК-3.2)
Ниже приведен код для работы с JSON-файлом. Завершите его, чтобы извлечь данные из JSON-файла и сохранить их в DataFrame с помощью библиотеки pandas.

python

```
import pandas as pd
import json

# Открываем JSON-файл и извлекаем данные
with open('data.json', 'r') as file:
```

```
    data = json.load(file)

# Создаем DataFrame из данных JSON
df = pd.DataFrame(data)

# Выводим DataFrame
print(df)
```

Задание 2 (ИОПК-6.1)

Напишите код для чтения данных из файла Excel и сохранения их в DataFrame с помощью библиотеки pandas.

```
python

import pandas as pd

# Чтение данных из файла Excel
df = pd.read_excel('data.xlsx')

# Выводим DataFrame
print(df)
```

Задание 3 (ИОПК-1.2)

Напишите код для парсинга HTML-страницы с помощью библиотеки BeautifulSoup и извлечения всех заголовков h1.

```
python

import requests
from bs4 import BeautifulSoup

# Выполнение GET-запроса к веб-странице
response = requests.get('https://example.com')

# Парсинг HTML-страницы
soup = BeautifulSoup(response.text, 'html.parser')

# Извлечение всех заголовков h1
headers = soup.find_all('h1')

# Выводим заголовки
for header in headers:
    print(header.text)
```

Задание 4 (ИПК-1.1)

Напишите код для предварительного анализа данных вакансий из DataFrame, включая вывод основных статистических характеристик.

```
python

import pandas as pd
```

```
# Загрузка данных из файла CSV в DataFrame
df = pd.read_csv('vacancies.csv')

# Предварительный анализ данных
print(df.describe())
```

Проект №1 (ИОПК-1.2, ИОПК-3.2, ИОПК-6.1, ИПК-1.1)

Создайте проект, включающий следующие этапы:

1. Извлечение данных из веб-источника или API.
2. Сохранение данных в базу данных.
3. Анализ данных с использованием SQL-запросов.
4. Презентация результатов анализа в виде отчета.

Пример выполнения проекта:
Извлечение данных из веб-источника или API
python

```
import requests
import pandas as pd

# Выполнение GET-запроса к API
response = requests.get('https://api.hh.ru/vacancies')
data = response.json()

# Преобразование данных в DataFrame
df = pd.DataFrame(data['items'])

# Выводим первые 5 строк DataFrame
print(df.head())
```

Сохранение данных в базу данных
python

```
import sqlite3

# Подключение к базе данных SQLite
conn = sqlite3.connect('vacancies.db')

# Сохранение данных в таблицу 'vacancies'
df.to_sql('vacancies', conn, if_exists='replace', index=False)

# Закрытие соединения
conn.close()
```

Анализ данных с использованием SQL-запросов
python

```
import sqlite3
import pandas as pd

# Подключение к базе данных SQLite
conn = sqlite3.connect('vacancies.db')
```

```
# SQL-запрос для анализа данных
query = """
SELECT
   name,
   COUNT(*) as vacancy_count
FROM
   vacancies
GROUP BY
   name
ORDER BY
   vacancy_count DESC
"""

# Выполнение запроса и создание DataFrame с результатами
result_df = pd.read_sql(query, conn)

# Закрытие соединения
conn.close()

# Вывод результатов анализа
print(result_df)
```

Презентация результатов анализа в виде отчета

```
python

import matplotlib.pyplot as plt

# Построение графика
plt.figure(figsize=(10, 6))
plt.bar(result_df['name'], result_df['vacancy_count'])
plt.xlabel('Job Titles')
plt.ylabel('Number of Vacancies')
plt.title('Number of Vacancies by Job Title')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Контрольная работа (ИОПК-1.2, ИОПК-3.2, ИОПК-6.1, ИПК-1.1)

Теоретические вопросы:

1. Опишите основные принципы работы с JSON в Python. (ИОПК-3.2)
2. Какие методы библиотеки requests используются для выполнения HTTP-запросов? Приведите примеры. (ИОПК-3.2)
3. Объясните, что такое агрегатные функции в SQL и приведите примеры их использования. (ИОПК-6.1)
4. Опишите процесс соединения таблиц в SQL и приведите пример. (ИОПК-6.1)

Задачи:

Задание 1 (ИОПК-6.1)

Напишите SQL-запрос для получения имен всех сотрудников, чья зарплата выше средней по компании.

sql

```sql
SELECT name
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

Задание 2 (ИОПК-3.2)
Напишите код для выполнения POST-запроса с данными JSON к API и сохранения ответа в DataFrame.

python

```python
import requests
import pandas as pd

# Данные для POST-запроса
data = {
   'key1': 'value1',
   'key2': 'value2'
}

# Выполнение POST-запроса
response = requests.post('https://api.example.com/data', json=data)

# Преобразование ответа в JSON и создание DataFrame
response_data = response.json()
df = pd.DataFrame(response_data)

# Выводим DataFrame
print(df)
```

Задание 3 (ИОПК-6.1)
Напишите SQL-запрос для получения данных из таблицы "employees" с сортировкой по возрасту в порядке убывания.

sql

```sql
SELECT * FROM employees ORDER BY age DESC;
```

Задание 4 (ИОПК-1.2)
Используйте библиотеку BeautifulSoup для извлечения всех ссылок (тегов a) с веб-страницы.

python

```python
import requests
from bs4 import BeautifulSoup

# Выполнение GET-запроса к веб-странице
response = requests.get('https://example.com')

# Парсинг HTML-страницы
```

```
soup = BeautifulSoup(response.text, 'html.parser')

# Извлечение всех ссылок (тегов `a`)
links = soup.find_all('a')

# Выводим все ссылки
for link in links:
    print(link.get('href'))
```

Задание 5 (ИОПК-6.1)

Напишите SQL-запрос для получения количества сотрудников в каждом отделе.

sql

```
SELECT department_id, COUNT(*) as employee_count
FROM employees
GROUP BY department_id;
```

Задание 6 (ИПК-1.1)

Напишите код для загрузки данных из CSV-файла в DataFrame и выполнения анализа данных, включая вывод уникальных значений в столбце.

python

```
import pandas as pd

# Загрузка данных из файла CSV в DataFrame
df = pd.read_csv('data.csv')

# Вывод уникальных значений в столбце
unique_values = df['column_name'].unique()

# Печать уникальных значений
print(unique_values)
```

Критерии оценивания:

Результаты контрольной работы определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценка «отлично» выставляется, если даны правильные ответы на все теоретические вопросы и все задачи решены без ошибок.

Оценка «хорошо» выставляется, если даны правильные ответы на не менее чем 3 из 4 теоретических вопросов, и допущено не более 1-2 незначительных ошибок в решении задач (не более 1 ошибки в каждой задаче).

Оценка «удовлетворительно» выставляется, если даны правильные ответы на не менее чем 2 из 4 теоретических вопросов, и допущено не более 3-4 незначительных ошибок в решении задач (не более 2 ошибок в каждой задаче).

Оценка «неудовлетворительно» выставляется, если даны правильные ответы менее чем на 2 теоретических вопроса и допущено более 4 ошибок в решении задач.

Примеры ошибок:

Незначительные ошибки:

Ошибки синтаксиса, которые легко исправить (например, забытый двоеточие в конце строки).

Неправильное использование методов или функций, не влияющее на общий результат.

Мелкие логические ошибки, которые можно быстро исправить.

Значительные ошибки:

Неправильное понимание задания, приводящее к неверному решению.

Ошибки, существенно влияющие на результат выполнения задачи.

Пропущенные важные шаги в решении задач.

Дополнительные критерии:

Отличное выполнение должно демонстрировать полное и глубокое понимание материала.

Хорошее выполнение должно показывать уверенное знание материала с небольшими недочетами.

Удовлетворительное выполнение показывает базовое понимание и способность решать основные задачи.

Неудовлетворительное выполнение указывает на необходимость дополнительного изучения материала и исправления пробелов в знаниях.

3. Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания

Промежуточная аттестация проводится в формате итогового индивидуального проекта. Индивидуальный проект представляет из себя письменный отчет (файл расширения docx/doc/pdf) на основе проведенного исследования, в рамках которого самостоятельно ставится исследовательский вопрос, выбираются данные, оценивается модель.

Отчет состоит из двух частей. Структура отчета следующая.

Часть 1

1. постановка задачи
2. ход исследования
3. обзор имеющегося опыта
4. формулировка ожидаемых результатов
5. данные и их подготовка
6. выбранная эконометрическая модель

Часть 2

1. описание данных, в т.ч. описательная статистика
2. оценка моделей, проведение тестов
3. интерпретация результатов
4. выводы

На основании оценок и комментариев преподавателя по первой части отчета, выполняется вторая часть отчета с учетом замечаний. Подробные требования к каждому пункту обозначаются преподавателем в презентации на первом занятии.

Экзамен оценивается по результатам текущего контроля – выполнение промежуточных тестов по темам курса (11 тестов= 66 баллам) и по результатам индивидуального итогового индивидуального проекта (34 балла).

Критерии оценивания итогового проекта:

Часть 1 (19 баллов)

1. постановка задачи (3)
2. ход исследования (2)
3. обзор имеющегося опыта (4)
4. формулировка ожидаемых результатов (2)
5. данные и их подготовка (4)
6. выбранная эконометрическая модель (4)

Часть 2 (15 баллов)

1. описание данных, в т.ч. описательная статистика (4)
2. оценка моделей, проведение тестов (5)
3. интерпретация результатов (3)
4. выводы (3)

На основании оценок и комментариев преподавателя по первой части отчета, выполняется вторая часть отчета с учетом замечаний.

Итоговая оценка складывается из суммы баллов, набранных при прохождении тестовых заданий, предусмотренных для текущего контроля и баллов, набранных при выполнение итогового проекта:

Итоговая сумма набранных баллов	**Оценка**
≤ 40	неудовлетворительно
от 41 до 60	удовлетворительно
от 61 до 80	хорошо
от 81 до 100	отлично

4. Оценочные материалы для проверки остаточных знаний (сформированности компетенций)

Теоретические вопросы

1. Опишите процесс работы с текстовыми файлами и файлами Excel в Python с использованием библиотеки pandas. (ИОПК-3.2)

2. Какие методы библиотеки BeautifulSoup используются для парсинга веб-страниц? Приведите примеры их использования. (ИОПК-3.2)

3. Что такое агрегатные функции в SQL и как они используются? Приведите примеры. (ИОПК-6.1)

4. Объясните, что такое соединение таблиц (JOIN) в SQL и какие типы соединений существуют. Приведите примеры. (ИОПК-6.1)

Тест

1. Какие из нижеперечисленных форматов файлов поддерживаются библиотекой pandas для чтения данных? (ИОПК-3.2)

- CSV

- Excel
- JSON
- XML

2. Какой SQL-запрос используется для фильтрации строк в таблице? (ИОПК-6.1)

- SELECT * FROM table;
- SELECT * FROM table WHERE condition;
- SELECT * FROM table ORDER BY column;
- SELECT * FROM table GROUP BY column;

3. Какие из следующих функций относятся к агрегатным функциям в SQL? (ИОПК-6.1)

- SUM()
- GROUP BY
- ORDER BY
- HAVING

4. Какой метод библиотеки requests используется для выполнения POST-запросов? (ИОПК-3.2)

- requests.get()
- requests.post()
- requests.put()
- requests.delete()

Задачи

Задача 1 (ИОПК-6.1)
Напишите SQL-запрос для получения имен всех сотрудников, работающих в отделе "Marketing", чья зарплата выше средней по отделу.

sql

```
SELECT name
FROM employees
WHERE department = 'Marketing' AND salary > (SELECT AVG(salary) FROM employees WHERE department = 'Marketing');
```

Задача 2 (ИОПК-3.2)
Напишите код для выполнения GET-запроса к API и сохранения ответа в JSON-файл.

python

```
import requests
import json

response = requests.get('https://api.example.com/data')
data = response.json()

with open('data.json', 'w') as file:
```

```
    json.dump(data, file)
```

Задача 3 (ИОПК-6.1)

Напишите SQL-запрос для получения данных из таблицы "products", отсортированных по цене в порядке убывания.

```
sql

SELECT * FROM products ORDER BY price DESC;
```

Задача 4 (ИОПК-1.2)

Используйте библиотеку BeautifulSoup для извлечения всех заголовков (тегов h1) с веб-страницы.

```
python

import requests
from bs4 import BeautifulSoup

response = requests.get('https://example.com')
soup = BeautifulSoup(response.text, 'html.parser')
headers = soup.find_all('h1')

for header in headers:
    print(header.text)
```

Задача 5 (ИОПК-6.1)

Напишите SQL-запрос для получения количества сотрудников в каждом отделе.

```
sql

SELECT department, COUNT(*) as employee_count
FROM employees
GROUP BY department;
```

Задача 6 (ИПК-1.1)

Напишите код для загрузки данных из Excel-файла в DataFrame и выполнения анализа данных, включая вывод средних значений в столбце "salary".

```
python

import pandas as pd

df = pd.read_excel('data.xlsx')
average_salary = df['salary'].mean()
print(f'Average Salary: {average_salary}')
```

Кейс

Описание кейса (ИОПК-1.2, ИОПК-6.1, ИПК-1.1)

Условие:

Вам необходимо создать систему для мониторинга вакансий на сайте HeadHunter. Система должна включать следующие компоненты:

Сбор данных о вакансиях с помощью API HeadHunter.
Сохранение данных в базу данных.
Анализ собранных данных для выявления трендов на рынке труда.
Решение:

Сбор данных
python

```
import requests

url = 'https://api.hh.ru/vacancies'
params = {
   'text': 'Data Scientist',
   'area': '1', # Москва
   'per_page': 100
}

response = requests.get(url, params=params)
vacancies = response.json()['items']
```

Сохранение данных
python

```
import sqlite3

conn = sqlite3.connect('hh_vacancies.db')
c = conn.cursor()
c.execute('''CREATE TABLE IF NOT EXISTS vacancies
             (id TEXT PRIMARY KEY, name TEXT, area TEXT, salary_from REAL, salary_to REAL, created_at TEXT)''')

for vacancy in vacancies:
   c.execute('INSERT OR IGNORE INTO vacancies (id, name, area, salary_from, salary_to, created_at) VALUES (?, ?, ?, ?, ?, ?)',
             (vacancy['id'], vacancy['name'], vacancy['area']['name'],
              vacancy['salary']['from'] if vacancy['salary'] else None,
              vacancy['salary']['to'] if vacancy['salary'] else None,
              vacancy['created_at']))

conn.commit()
conn.close()
```

Анализ данных
python

```
import pandas as pd
import sqlite3

conn = sqlite3.connect('hh_vacancies.db')
df = pd.read_sql_query('SELECT * FROM vacancies', conn)

# Пример анализа данных: Средняя зарплата по вакансиям
average_salary = df[['salary_from', 'salary_to']].mean()
```

```
print(f'Average Salary From: {average_salary["salary_from"]}')
print(f'Average Salary To: {average_salary["salary_to"]}')

# Пример анализа данных: Количество вакансий по месяцам
df['created_at'] = pd.to_datetime(df['created_at'])
df['month'] = df['created_at'].dt.to_period('M')
vacancies_per_month = df['month'].value_counts().sort_index()
print(vacancies_per_month)
```

Интерпретация выводов:

На основе собранных данных можно сделать выводы о востребованности профессии Data Scientist в Москве, проанализировать динамику изменения средней заработной платы по вакансиям и выявить сезонные тренды на рынке труда.

Информация о разработчиках

Салаева А.С., методист Skillfactory