«Рымбек Байсеитов атындағы Семей қаржы-экономикалық колледжі» КМҚК

**№26** С**абақ жоспары**

**Модуль /пән атауы:** Алгоритмдеу және бағдарламалау

**Дайындаған педагог:** Қалымжанова А.Б.

  2020 жылғы «10» қараша

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **Жалпы мәліметтер:** | | | |
| Курс, оқу жылы, топ: | | ІІІ курс, 2020-2021 оқу жылы, 18 IS-1 | |
| Сабақ типі | | аралас | |
| 1. **Мақсаты, міндеттер** | | | |
| Оқу сабақтары барысында білім алушылар игеретін кәсіби біліктердің тізбесі | | - Студенттерге Python бағдарламасында элементтерді орналастыруды үйрету: деректер қорын пайдалану, бағдарламада қолдану  - Деректер қорына арналған қосымшаларды құрастырады | |
| 1. **Сабақты жабдықтау** | | | |
| Оқу-әдістемелік құрал-жабдықтар, анықтамалық әдебиеттер | | Лутц М. Программирование на Python  Д. Ю. Федоров «Основы программирования на примере языка Python» | |
| Техникалық құралдар, материалдар | | Дербес компьютер, Python бағдарламасы | |
| 1. **Сабақтың барысы** | | | |
| 4.1. | **Ұйымдастыру**   1. Компьютерлердің сабаққа дайындығын тексеру 2. Оқушылармен амандасу 3. Оқушыларды түгендеу 4. Сабақтың мақсатымен таныстыру | | |
| 4.2. | **Үй тапсырмасын сұрау.**   1. №2 аралық бақылау жұмысын талдау | | |
| 4.3. | **Жаңа сабақ**  **Курс тақырыбы:** Бағандар бағанының негізгі алгаритмдерін қолданады  **Сабақ** **тақырыбы:** Берілгендер қорын құру | | |
| 4.4 | **Жаңа сабақты бекіту**  **Бекіту сұрақтары:**   1. DB-API 2 дегеніміз не? 2. Конструкторда қандай атаулы параметрлер болуы керек? 3. Транзакция дегеніміз не? 4. DB-API 2.0 деректер қорының өрістерін сипаттау үшін қандай нысандар-типтері қарастырылған? | | |
| 4.5. | **Студенттерді бағалау.** | | |
| 1. **Сабақ бойынша рефлексия** | | | Онлайн чатқа смайлик арқылы немесе келесі сөздер арқылы сабаққа деген көзқарас:  белсенді / немғұрайлы  Көңілім толды / толмады  ұзақ / қысқа немесе тез  шаршамадым / шаршадым  жақсарды / төмендеді  түсінікті / түсініксіз  пайдалы / пайдасыз  Қызықты / қызықсыз |
| 1. **Үй тапсырмасы** | | | Лутц М. Программирование на Python 150-155 беттер |

**Дәріс №26**

**Сабақ тақырыбы: Берілгендер қорын құру**

1. Реляциялық ДҚБЖ негізгі ұғымдары.

Реляциялық деректер қоры-бұл деректер кестесі.

Кесте-жолдар мен бағандардан тұратын тікбұрышты матрица. Кесте (relation) қатынасын көрсетеді.

Жол-баған жолдарынан тұратын жазба. Әрбір өрісте кейбір мән немесе арнайы NULL (бос) мәні болуы мүмкін. Кестеде еркін жолдар саны болуы мүмкін. Реляциялық модель үшін жолдардың орналасу тәртібі анықталмаған және маңызды емес.

Кестедегі әр бағанның өз аты мен түрі бар.

DB-API 2 дегеніміз не.

Атауына енгізілген аббревиатуралар екі ұғымды біріктіреді: DB (Database, деректер қоры) және API (Application Program Interface, қолданбалы бағдарламаның интерфейсі).

Осылайша, DB-API деректер базасымен қолданбалы бағдарламаның интерфейсін анықтайды. Бұл интерфейс төменде сипатталған, Python-бағдарламалардың деректер базасымен байланысы үшін қызмет ететін барлық кеңею модульдерін іске асыруы тиіс. Бірыңғай API (қазіргі уақытта оның екінші нұсқасы) пайдаланылатын деректер базасының маркасынан абстракциялауға мүмкіндік береді, Қажет болған жағдайда функциялар мен әдістердің бір жиынтығын зерттей отырып, бір ДББЖ-ны екіншісіне оңай ауыстыруға мүмкіндік береді.

DB-API 2.0 PEP 249-да сипатталған (сайт http://www.python.org/dev/peps/pep-0249/) және төменде берілген сипаттама оған негізделген.

Сипаттамасы DB API 2.0

DB API 2.0 деректер базасымен жұмыс істеу үшін кеңейту Модулінің интерфейстерін, объект-базамен қосылу әдістерін, ағымдағы өңделетін жазбаның объект-курсорын, деректер типтері мен олардың конструкторларына арналған әр түрлі объектілерді регламенттейді, сондай-ақ модульдерді іске асыру бойынша әзірлеушілерге арналған ұсынымдарды қамтиды. Бүгінгі күні Python көптеген белгілі деректер базасын кеңейту модульдері арқылы қолдайды (мекен-жай бойынша web-бетте нақтылауға болады) http://www.python.org/topics/database/). төменде жаңа модульдерді әзірлеушілерге арналған ұсынымдарды қоспағанда, DB-API-дің барлық дерлік ережелері қарастырылады.

2. Модуль интерфейсі.

Мұнда DB-API 2.0 талаптарын қанағаттандыру үшін модульді ұсыну керек.

Деректер қорына кіру қосылым объектісі (connection object) арқылы жүзеге асырылады. DB-API-үйлесімді модуль қосылымдар класы үшін connect() конструктор функциясын ұсынуы тиіс. Конструкторда келесі атаулы параметрлер болуы керек:

* dsn деректер көзінің атауы
* user пайдаланушы аты
* Password Пароль
* host ДББЖ жұмыс істейтін хост Мекен-жайы
* database деректер қорының атауы.
* Нысан әдістері-қосылыстар кейінірек қарастырылады.

Модуль оның негізгі сипаттамалары бар константаларды анықтайды:

* apilevel нұсқасы DB-API ("1.0" немесе "2.0").
* threadsafety басқару ағындарын пайдалану кезінде модуль мүмкіндіктерін сипаттайтын бүтін тұрақты:
* 0 Модуль ағындарды қолдамайды.
* 1 ағындар модульді бірге пайдалана алады, бірақ қосылыстар емес.
* 2 ағындар модуль мен қосылыстарды бірлесіп пайдалана алады.
* 3 ағындар модульді, қосылымдарды және меңзерлерді бірлесіп пайдалана алады. (Бірлесіп пайдалану деп аталған ресурстарды семафорларды қолданбай пайдалану мүмкіндігі түсініледі).
* paramstyle параметрлерді қойған кезде қолданылатын белгі түрі. Осы константаның келесі мәндері мүмкін:
* "format" ANSI C стилінде пішімдеу (мысалы, "% s", "% i").
* "pyformat "Python ("%(item)s стилінде пішімде аталған спецификаторларды пайдалану" )
* "qmark" белгілерді пайдалану "?"параметрлерді орнату орындарын белгілеу үшін.
* "numeric" позициялар нөмірлерін пайдалану ( ": 1").
* "named "орнатылған параметрлердің атын пайдалану (": name").

Модуль типтік ерекше жағдайларды белгілеу үшін бірқатар ерекшеліктерді анықтауы тиіс: Warning (ескерту), Error (Қате), InterfaceError (интерфейс қатесі), DatabaseError (деректер базасына қатысты қате). Сондай-ақ, осы соңғы ерекшеліктің Ішкі сыныбы: DataError (деректерді өңдеу қатесі), OperationalError (жұмыс қатесі немесе деректер қоры қатесі), IntegrityError (деректер қоры тұтастығының қатесі), InternalError (деректер қорының ішкі қатесі), ProgrammingError (бағдарламалық қате, мысалы, SQL-сұранымның синтаксисіндегі қате), NotSupportedError (сұралған сипаттың қолдауы болмаған жағдайда).

Нысан-қосылым

Объект-connect () функциясын сәтті шақыру нәтижесінде алынған қосылым келесі әдістерге ие болуы тиіс:

* close () деректер базасымен байланысты жабады.
* commit () транзакцияны аяқтайды.
* rollback () басталған транзакция (бастапқы күйін қалпына келтіреді). Аяқталмаған транзакция кезінде байланысты жабу транзакцияны автоматты түрде қайтарады.
* cursor () осы қосылымды пайдаланатын курсорды қайтарады. Егер деректер базасы меңзерді қолдамаса, жұптастыру модулі оларды имитациялауы тиіс.

Транзакция деп Деректер базасын өзгертетін бір немесе бірнеше операциялардан тұратын топ түсініледі. Транзакция деректер қорына логикалық бөлінбейтін операцияларға сәйкес келеді, ал транзакцияны ішінара орындау ДБ тұтастығын бұзуға әкеледі. Мысалы, бірінші шотты азайту және екіншісін ұлғайту бойынша бір шоттан басқа операцияларға ақша аудару кезінде транзакция болып табылады. Commit() және rollback() әдістері транзакцияның басталуы мен аяқталуын айқын көрсетеді. Айтпақшы, барлық деректер қоры транзакция механизмін қолдамайды.

Модульдің DB-API 2.0 іске асырылуына байланысты осы қосылыс курсорларының жұмысын жалғастыратын объект-қосылысқа сілтемені сақтау қажет. Атап айтқанда, бұл нысан-курсорды бірден алуға болмайтынын білдіреді. Сондай-ақ, элементті-қосылымды жергілікті айнымалы функциядан немесе нысан-меңзерден қайтару арқылы қалдыруға болмайды.

Объект-курсор

Курсор (ағылш . cursor-current Set Of Records, ағымдағы жазбалар жиынтығы) сұрау нәтижесімен жұмыс істеу үшін қызмет етеді. Сұраныс нәтижесі әдетте бір немесе бірнеше тікбұрышты кестелер-өрістер және жол-жазбалары бар. Қолданба алынған кестелер мен жазбаларды бір кестеден оқи және өңдей алады, сондықтан курсорда ағымдағы кесте мен жазбалар туралы ақпарат сақталады. Кез келген уақытта белгілі бір курсор бір SQL-нұсқаулықты орындауға байланысты.

Курсордың атрибуттары да DB-API анықталған:

arraysize Fetchmany () әдісімен қайтарылатын жазбалар санына тең Атрибут. Әдепкі бойынша 1 тең.

callproc (procname[, params]) сақталатын procname процедурасын params тізбегінің өзгермелі параметрлерімен шақырады. Сақталған рәсім кейбір реттілік параметрлерінің мәнін өзгерте алады. Әдіс fetch-әдістері арқылы қол жеткізу нәтижесін қайтаруы мүмкін.

close () курсорды жабады.

description бұл тек оқу үшін қол жетімді атрибут-семиэлементтік тізбектерден тізбектілік. Осы тізбектердің әрқайсысы нәтиженің бір бағанын сипаттайтын ақпаратты қамтиды:

(name, type\_code, display\_size, internal\_size, precision, scale, null\_ok)

Алғашқы екі элемент (аты және түрі) міндетті, ал қалғандарының орнына (шығару өлшемі, ішкі өлшемі, дәлдігі, масштабы, бос мәнді орындау мүмкіндігі) None мәні болуы мүмкін. Бұл атрибут мәндерді қайтармайтын операциялар үшін None тең болуы мүмкін.

execute (operation[, parameters]) деректер базасына немесе ДББЖ командасын сұранымды орындайды. Параметрлер (parameters ) деректер базасында жоғарыда сипатталған paramstyle атрибуттарына сәйкес ұсынылуы мүмкін.

executemany (operation, seq\_of\_parameters) берілген үлгіге параметрлерді орната отырып, сұраулар немесе командалар сериясын орындайды. Seq\_of\_parameters параметрі параметрлер жиынтықтарының ретін анықтайды.

fetchall () сұрау нәтижесінің барлық (немесе барлық қалған) жазбаларын қайтарады.

fetchmany ([size]) сұрау нәтижесінен келесі бірнеше жазбаларды кезектілік ретімен қайтарады. Бос бірізділік деректер жоқ дегенді білдіреді. Қосымша size параметрі қайтарылатын жазбалар санын көрсетеді (нақты қайтарылатын жазбалар аз болуы мүмкін). Әдепкі size курсордың arraysize атрибутына тең.

fetchone () деректер болмаған кезде сұрау нәтижесінен немесе None келесі жазбаны (ретпен) қайтарады.

nextset () курсорды сұрау нәтижесінде алынған келесі деректер жиынтығының басына ауыстырады (бұл ретте алдыңғы жиынтықтағы жазбалардың бөлігі оқылмаған күйінде қалуы мүмкін). Егер жиынтық болмаса, None қайтарады. Барлық деректер қоры бір операция үшін бірнеше нәтиже жиынтығын қайтаруды қолдамайды.

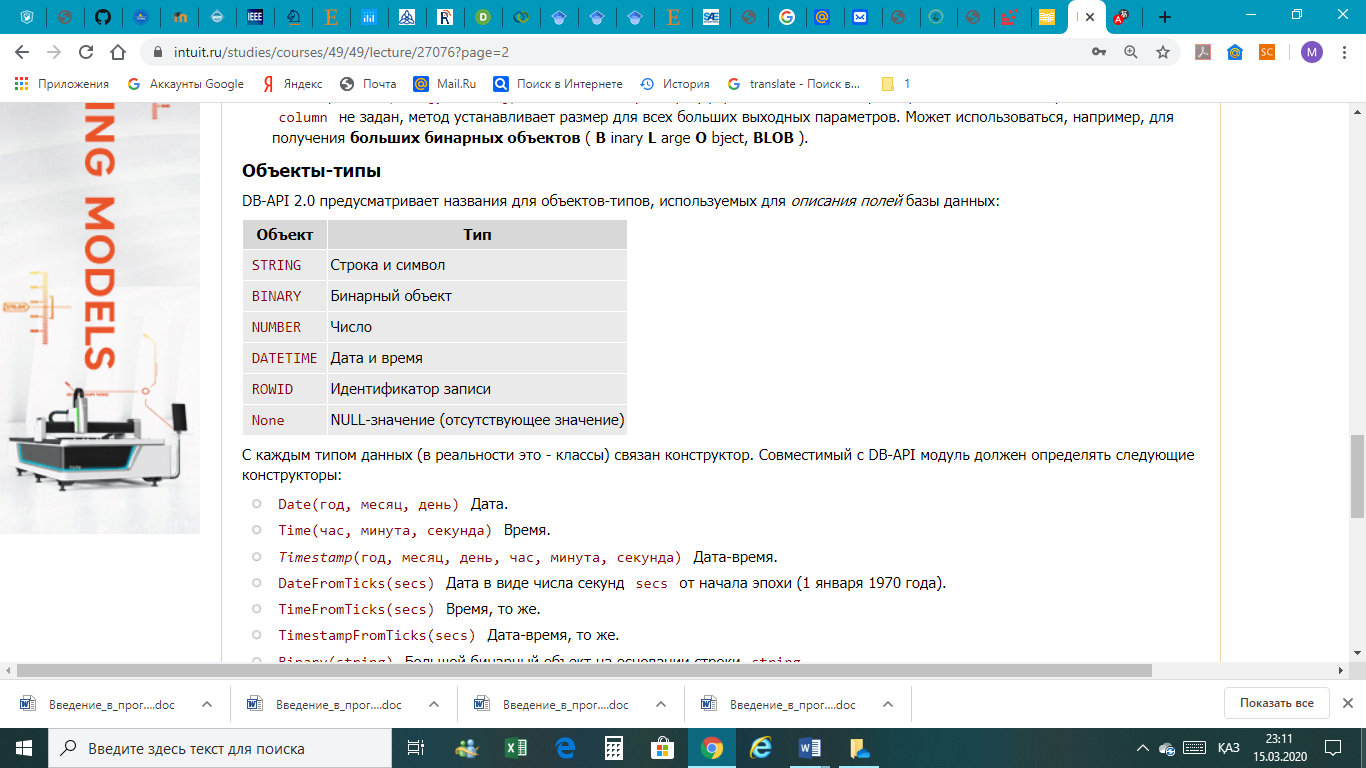
rowcount соңғы сұрау нәтижесінде алынған немесе қозғалған жазбалар саны. Execute сұраулары болмаған немесе жазбалардың санын көрсету мүмкін болмаған жағдайда -1 тең.

setinputsizes (sizes) операцияларда пайдаланылатын параметрлер үшін жад аймағын алдын ала анықтайды. Sizes аргументі әрбір элемент бір кіріс параметріне сәйкес келетін тізбекті анықтайды. Элемент тиісті параметрдің түрі немесе жолдың ұзындығын білдіретін бүтін сан болуы мүмкін. Егер кіріс параметрінің мөлшері туралы ештеңе алдын ала айтылмаса немесе ол өте үлкен деп болжанса, ол сондай-ақ None мәні болуы мүмкін. Әдіс execute әдістеріне дейін туындауы тиіс.

setoutputsize (size[, column]) column нөмірі бар бағаннан шығу параметрі үшін буфер өлшемін орнатады. Егер column орнатылмаса, әдіс барлық Үлкен Шығыс параметрлеріне өлшем орнатады. Мысалы, үлкен екілік нысандарды ( B inary L arge O bject, BLOB) алу үшін пайдаланылуы мүмкін.

Нысандар-түрлері

DB-API 2.0 деректер қорының өрістерін сипаттау үшін пайдаланылатын нысандар-типтері үшін атауларды қарастырады:



Әрбір деректер түріне (шын мәнінде бұл - сыныптар) конструктор байланысты. DB-API үйлесімді модуль келесі конструкторларды анықтауы тиіс:

* Date(год, месяц, день) Дата.
* Time(час, минута, секунда) Время.
* Timestamp(год, месяц, день, час, минута, секунда) Дата-время.
* DateFromTicks(secs) Дата в виде числа секунд secs от начала эпохи (1 января 1970 года).
* TimeFromTicks(secs) Время, то же.
* TimestampFromTicks(secs) Дата-время, то же.
* Binary(string) Большой бинарный объект на основании строки string.