



VERİ TABANI YÖNETİM SİSTEMLERİ

PROJE RAPORU

Öğrenci Bilgileri:

G221210013

Berke Pite

berke.pite@ogr.sakarya.edu.tr

[Github](#)

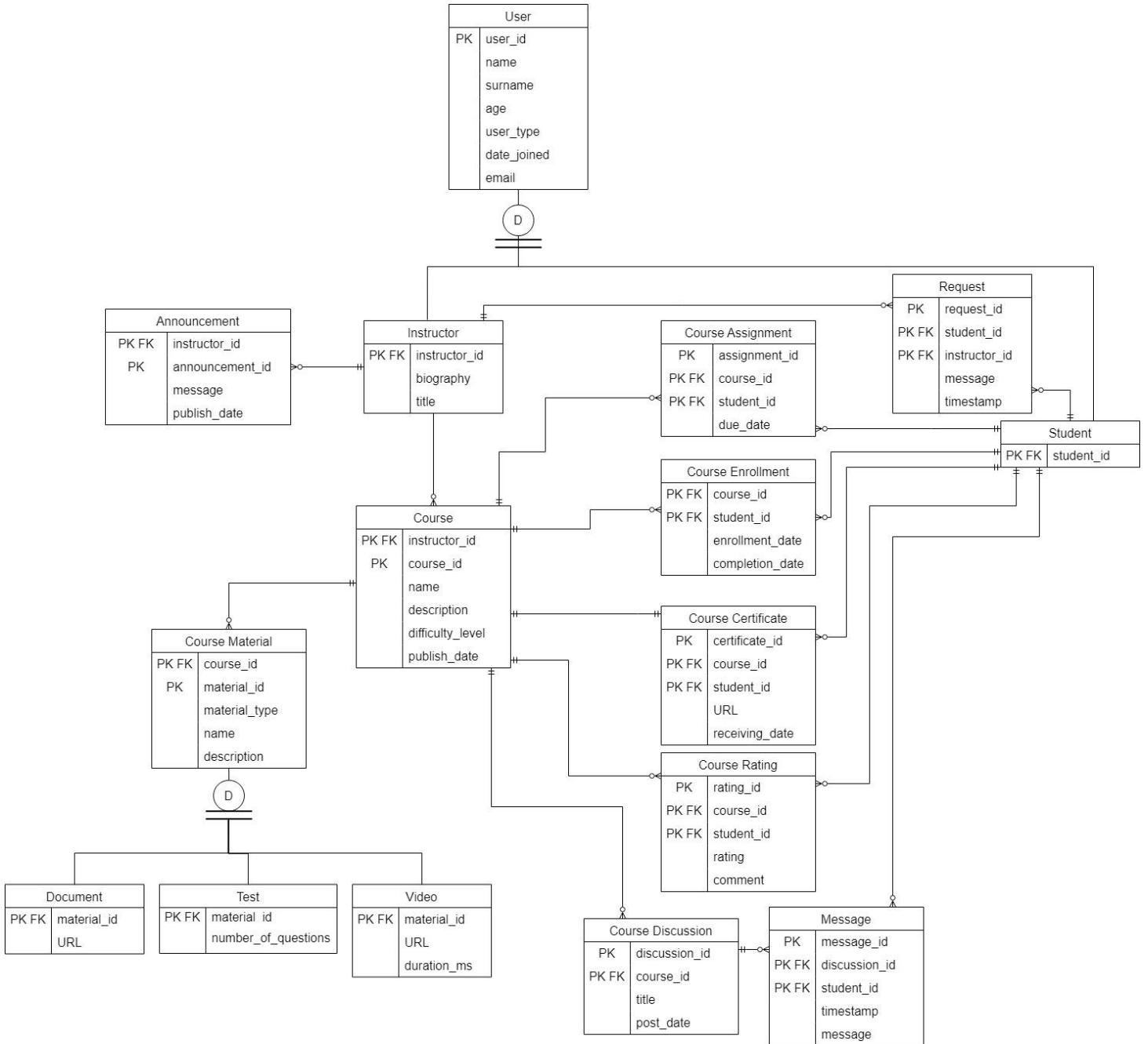
Uygulama:

Online Eğitim Platformu, kullanıcıların eğitmen veya öğrenci olarak katılmasına izin veren, interaktif bir öğrenme ortamı sunan bir platformdur. Eğitmenler kendi kurslarını oluşturabilir, düzenleyebilir ve paylaşabilir; öğrencilere çeşitli görevler vererek etkileşimli öğrenme deneyimi sağlar. Kurslar, tartışma forumları ve farklı zorluk seviyeleri ile çeşitli öğrenme materyallerini içerir. Eğitmenler duyuru yapabilir ve öğrencileri puanlayabilir; öğrenciler ise katıldıkları kursları değerlendirerek sertifika alma hakkı elde eder. İki yönlü etkileşimle öğrenci ve eğitmenler arasında dinamik bir öğrenme ekosistemi oluşturur.

İş Kuralları:

- Online Eğitim Platformunun çok sayıda kullanıcısı vardır.
- Platformun eğitmenleri olabilir.
- Eğitmenler de bir kullanıcıdır.
- Bir kullanıcı hem eğitmen hem öğrenci olamaz.
- Eğitmenler kendi kurslarını paylaşabilir, silebilir, düzenleyebilir.
- Kullanıcılar öğrenci olup kurslarda yer alabilir.
- Kurslar kendi tartışma forumlarına sahiptir.
- Kurslar farklı zorluk seviyelerine sahiptir.
- Kursun tartışma forumlarında öğrenciler mesaj gönderebilir.
- Kurslar çeşitli öğrenme materyallerine sahip olabilir (Videolar, Testler, Dokümanlar).
- Eğitmenler öğrencilere çeşitli görevler verebilirler (Quiz, Sınav, Ödev).
- Eğitmenler duyuru yayınlayabilirler.
- Öğrenciler katılıp bitirdikleri kursları puanlayabilirler.
- Kursları bitiren öğrenciler kursun sertifikasını almaya hak kazanır.
- Öğrenciler eğitmenlere istekte bulunabilirler.

VARLIK BAĞINTI MODELİ:



İLİŞKİSEL ŞEMA:

user(user_id: integer, name: string, surname: string, user_type: character, date_joined: DATE, email: string)

instructor(instructor_id: integer, biography: string, title: string)

student(student_id: integer)

announcement(instructor_id: integer, announcement_id: integer, message: string, publish_date: date)

course(instructor_id: integer, course_id: integer, name: string, description: string, difficulty_level: character, publish_date: date)

course_material(course_id: integer, material_id: integer, material_type: character(3), name: string, description: string)

document(material_id: integer, URL: string)

test(material_id: integer, number_of_questions: integer)

video(material_id: integer, URL: string, duration_ms: integer)

request(request_id: integer, student_id: integer, instructor_id: integer, message: string, timestamp: timestamp)

assignment(assignment_id: integer, course_id: integer, student_id: integer, due_date: date)

enrollment(course_id: integer, student_id: integer, enrollment_date: date, completion_date: date)

certificate(certificate_id: integer, course_id: integer, student_id: integer, URL: string, receiving_date: date)

rating(course_id: integer, student_id: integer, rating: integer, comment: string)

discussion(discussion_id: integer, course_id: integer, title: string, post_date: date)

message(message_id: integer, discussion_id: integer, student_id: integer, timestamp: timestamp, message: string)

Tablo Oluşturma SQL Komutları:

```
CREATE TABLE public.announcement
```

```
(  
    announcement_id SERIAL,  
    instructor_id integer NOT NULL,  
    message character varying(1000),  
    publish_date date NOT NULL,  
    CONSTRAINT "announcementPK" PRIMARY KEY (announcement_id, instructor_id),  
    CONSTRAINT "instructorFK" FOREIGN KEY (instructor_id)REFERENCES "user".instructor (user_id)  
)
```

```
CREATE TABLE public.assignment
```

```
(  
    assignment_id SERIAL,  
    course_id integer NOT NULL,  
    student_id integer NOT NULL,  
    due_date date NOT NULL,  
    CONSTRAINT "assignmentPK" PRIMARY KEY (assignment_id, student_id, course_id),  
    CONSTRAINT "courseFK" FOREIGN KEY (course_id)REFERENCES public.course (course_id),  
    CONSTRAINT "studentFK" FOREIGN KEY (student_id)REFERENCES "user".student (user_id)  
)
```

```
CREATE TABLE public.certificate
```

```
(  
    certificate_id SERIAL,  
    course_id integer NOT NULL,  
    student_id integer NOT NULL,  
    receiving_date date NOT NULL,  
    "URL" character varying(150) NOT NULL,  
    CONSTRAINT "certificatePK" PRIMARY KEY (certificate_id, student_id, course_id),  
    CONSTRAINT "courseFK" FOREIGN KEY (course_id)REFERENCES public.course (course_id),  
    CONSTRAINT "studentFK" FOREIGN KEY (student_id)REFERENCES "user".student (user_id)  
)
```

```

CREATE TABLE public.course
(
    instructor_id integer NOT NULL,
    course_id SERIAL,
    name character varying(40) NOT NULL,
    description character varying(1000) NOT NULL,
    difficulty_level character(1),
    publish_date date NOT NULL,
    CONSTRAINT "coursePK" PRIMARY KEY (course_id, instructor_id),
    CONSTRAINT course_id_unique UNIQUE (course_id),
    CONSTRAINT "instructor_idFK" FOREIGN KEY (instructor_id)REFERENCES "user".instructor (user_id)
)

```

```

CREATE TABLE public.discussion
(
    discussion_id SERIAL,
    course_id integer NOT NULL,
    title character varying NOT NULL,
    post_date date NOT NULL,
    CONSTRAINT "discussionPK" PRIMARY KEY (discussion_id, course_id),
    CONSTRAINT discussion_id_unique UNIQUE (discussion_id),
    CONSTRAINT "courseFK" FOREIGN KEY (course_id)REFERENCES public.course (course_id)
)

```

```

CREATE TABLE public.enrollment
(
    course_id integer NOT NULL,
    student_id integer NOT NULL,
    enrollment_date date NOT NULL,
    completion_date date,
    CONSTRAINT "enrollmentPK" PRIMARY KEY (student_id, course_id),
    CONSTRAINT "courseFK" FOREIGN KEY (course_id)REFERENCES public.course (course_id),
    CONSTRAINT "studentFK" FOREIGN KEY (student_id)REFERENCES "user".student (user_id)
)

```

```

CREATE TABLE public.message
(
    message_id SERIAL,
    discussion_id integer NOT NULL,
    student_id integer NOT NULL,
    "timestamp" timestamp without time zone NOT NULL,
    message character varying(150) NOT NULL,
    CONSTRAINT "messagePK" PRIMARY KEY (message_id, discussion_id, student_id),
    CONSTRAINT "discussionFK" FOREIGN KEY (discussion_id)REFERENCES public.discussion (discussion_id),
    CONSTRAINT "studentFK" FOREIGN KEY (student_id)REFERENCES "user".student (user_id)
)

CREATE TABLE public.rating
(
    rating_id SERIAL,
    course_id integer NOT NULL,
    student_id integer NOT NULL,
    comment character varying(150),
    rating smallint NOT NULL,
    CONSTRAINT "ratingPK" PRIMARY KEY (rating_id, student_id, course_id),
    CONSTRAINT "courseFK" FOREIGN KEY (course_id)REFERENCES public.course (course_id),
    CONSTRAINT "studentFK" FOREIGN KEY (student_id)REFERENCES "user".student (user_id),
    CONSTRAINT rating_rating_check CHECK (rating >= 0 AND rating <= 10)
)

CREATE TABLE public.request
(
    request_id SERIAL,
    instructor_id integer NOT NULL,
    student_id integer NOT NULL,
    "timestamp" timestamp without time zone NOT NULL,
    message character varying(300) NOT NULL,
    CONSTRAINT "requestPK" PRIMARY KEY (request_id, instructor_id, student_id),
    CONSTRAINT "instructorFK" FOREIGN KEY (instructor_id)REFERENCES "user".instructor (user_id),
    CONSTRAINT "studentFK" FOREIGN KEY (student_id)REFERENCES "user".student (user_id))

```

Tabloya Veri Ekleme SQL Komutları:

Tabloya veri eklemek için kullanılan SQL komutları [github](#) adresinde bulunmaktadır.

Triggerlar:

`check_course_id()` -> bu trigger `course_id` sütunu gerektiren herhangi bir tabloya uygulanıp girilen `course_id`'nin var olup olmadığını kontrol eder, eğer yoksa hata döndürür.

`check_discussion()` -> bu trigger `discussion_id` sütunu gerektiren herhangi bir tabloya uygulanıp girilen `discussion_id`'nin var olup olmadığını kontrol eder, eğer yoksa hata döndürür.

`check_instructor()` -> bu trigger `instructor_id` sütunu gerektiren herhangi bir tabloya uygulanıp girilen `instructor_id`'nin var olup olmadığını kontrol eder, eğer yoksa hata döndürür.

`check_student()` -> bu trigger `student_id` sütunu gerektiren herhangi bir tabloya uygulanıp girilen `student_id`'nin var olup olmadığını kontrol eder, eğer yoksa hata döndürür.

Fonksiyonlar:

`calculate_average_rating(int course_id)` -> bu fonksiyon girilen `course_id` değeri için `rating` tablosuyla `course` tablosunu `course_id` üzerinden karşılaştırır ve öğrencilerin `rating`'lerinin ortalamasını döndürür.


```
insert_row(std::string row_data, PGconn *conn, std::string table_name,
          std::string table_columns)
```

fonksiyonu öncesi sonrası:

	announcement_id [PK] integer	instructor_id [PK] integer	message character varying (1000)
1	1	1	Fusce posuere felis sed lacus. Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue
2	2	2	Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue, risus semper porta volutpat, c
3	3	4	In eleifend quam a odio. In hac habitasse platea dictumst. Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut
4	4	5	Fusce congue, diam id ornare imperdiet, sapien urna pretium nisl, ut volutpat sapien arcu sed augue. Aliquam erat volutpat. In congue. Etiam justo. f
5	5	6	Curabitur gravida nisi at nibh. In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem. Integer
6	6	8	Donec ut dolor. Morbi vel lectus in quam fringilla rhoncus. Mauris enim leo, rhoncus sed, vestibulum sit amet, cursus id, turpis. Integer aliquet, mass
7	7	10	Morbi a ipsum. Integer a nibh. In quis justo. Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet. Maecenas leo odio, condim
8	8	13	Duis mattis egestas metus. Aenean fermentum. Donec ut mauris eget massa tempor convallis. Nulla neque libero, convallis eget, eleifend luctus, ult
9	9	18	Vestibulum quam sapien, varius ut, blandit non, interdum in, ante. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Cu
10	10	15	Ut tellus. Nulla ut erat id mauris vulputate elementum. Nullam varius. Nulla facilisi. Cras non velit nec nisi vulputate nonummy. Maecenas tincidunt l

	announcement_id [PK] integer	instructor_id [PK] integer	message character varying (1000)
1	1	1	Fusce posuere felis sed lacus. Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congu
2	2	2	Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue, risus semper porta volutpat,
3	3	4	In eleifend quam a odio. In hac habitasse platea dictumst. Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut
4	4	5	Fusce congue, diam id ornare imperdiet, sapien urna pretium nisl, ut volutpat sapien arcu sed augue. Aliquam erat volutpat. In congue. Etiam justo.
5	5	6	Curabitur gravida nisi at nibh. In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem. Integer
6	6	8	Donec ut dolor. Morbi vel lectus in quam fringilla rhoncus. Mauris enim leo, rhoncus sed, vestibulum sit amet, cursus id, turpis. Integer aliquet, mas
7	7	10	Morbi a ipsum. Integer a nibh. In quis justo. Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet. Maecenas leo odio, condit
8	8	13	Duis mattis egestas metus. Aenean fermentum. Donec ut mauris eget massa tempor convallis. Nulla neque libero, convallis eget, eleifend luctus, ul
9	9	18	Vestibulum quam sapien, varius ut, blandit non, interdum in, ante. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia C
10	10	15	Ut tellus. Nulla ut erat id mauris vulputate elementum. Nullam varius. Nulla facilisi. Cras non velit nec nisi vulputate nonummy. Maecenas tincidunt
11	11	1	Bugün kurs yayınlanmayacaktır!

```
void insert_row(std::string row_data, PGconn *conn, std::string table_name,
               std::string table_columns) {
    std::string query = "insert into ";
    query.append(str: table_name);
    query.append(s: " (");
    query.append(str: table_columns);
    query.append(s: ") ");
    query.append(s: "values (");
    query.append(str: row_data);
    query.append(s: ")");

    PGresult *res = PQexec(conn, query.c_str());
    std::cout << PQerrorMessage(conn);
}
```

print_table(const std::string &table) fonksiyonu çıktısı:

```
rating_id--course_id--student_id--comment--rating;
1--1--3--Aliquam non mauris. Morbi non lectus.--1;
2--2--32--Donec semper sapien a libero. Nam dui.--2;
3--3--58--Phasellus sit amet erat. Nulla tempus.--6;
4--4--11--In congue. Etiam justo.--4;
5--5--61--Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet.--5;
6--6--14--Donec quis orci eget orci vehicula condimentum. Curabitur in libero ut massa volutpat convallis.--4;
7--7--95--Integer aliquet, massa id lobortis convallis, tortor risus dapibus augue, vel accumsan tellus nisi eu orci. Mauris lacinia sapien quis libero.--2;
8--8--17--Nullam porttitor lacus at turpis. Donec posuere metus vitae ipsum.--0;
9--9--43--Donec odio justo, sollicitudin ut, suscipit a, feugiat et, eros. Vestibulum ac est lacinia nisi venenatis tristique.--4;
10--10--21--Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae;
Duis faucibus accumsan odio. Curabitur convallis.--10;
11--11--99--Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae;
Mauris viverra diam vitae quam. Suspendisse potenti.--2;
12--12--48--Integer non velit. Donec diam neque, vestibulum eget, vulputate ut, ultrices vel, augue.--9;
13--13--49--Duis bibendum, felis sed interdum venenatis, turpis enim blandit mi, in porttitor pede justo eu massa. Donec dapibus.--5;
14--14--77--Pellentesque at nulla. Suspendisse potenti.--6;
15--15--51--Integer ac leo. Pellentesque ultrices mattis odio.--10;
```

get_table(std::string table_name, PGconn *conn)

fonksiyonu: Bu fonksiyon tablodaki bütün verileri bir string'e atar ve ve print table için yazılması uygun hale getirir.

delete_row(PGconn *conn, std::string table_name, std::string condition) fonksiyonu öncesi ve sonrası:

	announcement_id [PK] integer	instructor_id [PK] integer	message character varying (1000)
1	1	1	Fusce posuere felis sed lacus. Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue
2	2	2	Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue, risus semper porta volutpat,
3	3	4	In eleifend quam a odio. In hac habitasse platea dictumst. Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut
4	4	5	Fusce congue, diam id ornare imperdiet, sapien urna pretium nisl, ut volutpat sapien arcu sed augue. Aliquam erat volutpat. In congue. Etiam justo.
5	5	6	Curabitur gravida nisi at nibh. In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem. Integer
6	6	8	Donec ut dolor. Morbi vel lectus in quam fringilla rhoncus. Mauris enim leo, rhoncus sed, vestibulum sit amet, cursus id, turpis. Integer aliquet, mas
7	7	10	Morbi a ipsum. Integer a nibh. In quis justo. Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet. Maecenas leo odio, condir
8	8	13	Duis mattis egestas metus. Aenean fermentum. Donec ut mauris eget massa tempor convallis. Nulla neque libero, convallis eget, eleifend luctus, ul
9	9	18	Vestibulum quam sapien, varius ut, blandit non, interdum in, ante. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia C
10	10	15	Ut tellus. Nulla ut erat id mauris vulputate elementum. Nullam varius. Nulla facilisi. Cras non velit nec nisi vulputate nonummy. Maecenas tincidunt
11	11	1	Bugün kurs yayınlanmayacaktır!

	announcement_id [PK] integer	instructor_id [PK] integer	message character varying (1000)
1	1	1	Fusce posuere felis sed lacus. Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue
2	2	2	Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue, risus semper porta volutpat, c
3	3	4	In eleifend quam a odio. In hac habitasse platea dictumst. Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut
4	4	5	Fusce congue, diam id ornare imperdiet, sapien urna pretium nisl, ut volutpat sapien arcu sed augue. Aliquam erat volutpat. In congue. Etiam justo. f
5	5	6	Curabitur gravida nisi at nibh. In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem. Integer
6	6	8	Donec ut dolor. Morbi vel lectus in quam fringilla rhoncus. Mauris enim leo, rhoncus sed, vestibulum sit amet, cursus id, turpis. Integer aliquet, mass
7	7	10	Morbi a ipsum. Integer a nibh. In quis justo. Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet. Maecenas leo odio, condim
8	8	13	Duis mattis egestas metus. Aenean fermentum. Donec ut mauris eget massa tempor convallis. Nulla neque libero, convallis eget, eleifend luctus, ult
9	9	18	Vestibulum quam sapien, varius ut, blandit non, interdum in, ante. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Cu
10	10	15	Ut tellus. Nulla ut erat id mauris vulputate elementum. Nullam varius. Nulla facilisi. Cras non velit nec nisi vulputate nonummy. Maecenas tincidunt l

```

void delete_row(PGconn *conn, std::string table_name, std::string condition) {
    std::string query = "delete from ";
    query.append(str: table_name);
    query.append(s: " where ");
    query.append(str: condition);
    query.append(s: ";");

    PGresult *res = PQexec(conn, query: query.c_str());
    std::cout << PQerrorMessage(conn);
}

```

update_row(PGconn *conn, std::string table_name, std::string column, std::string condition, std::string data) fonksiyonu öncesi ve sonrası:

	announcement_id [PK] integer	instructor_id [PK] integer	message character varying (1000)
1		1	Fusce posuere felis sed lacus. Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue
2		2	Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue, risus semper porta volutpat,
3		3	In eleifend quam a odio. In hac habitasse platea dictumst. Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut
4		4	Fusce congue, diam id ornare imperdiet, sapien urna pretium nisl, ut volutpat sapien arcu sed augue. Aliquam erat volutpat. In congue. Etiam justo.
5		5	Curabitur gravida nisi at nibh. In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem. Integer
6		6	Donec ut dolor. Morbi vel lectus in quam fringilla rhoncus. Mauris enim leo, rhoncus sed, vestibulum sit amet, cursus id, turpis. Integer aliquet, mas
7		7	Morbi a ipsum. Integer a nibh. In quis justo. Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet. Maecenas leo odio, condit
8		8	Duis mattis egestas metus. Aenean fermentum. Donec ut mauris eget massa tempor convallis. Nulla neque libero, convallis eget, eleifend luctus, ul
9		9	Vestibulum quam sapien, varius ut, blandit non, interdum in, ante. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia C
10		10	Ut tellus. Nulla ut erat id mauris vulputate elementum. Nullam varius. Nulla facilisi. Cras non velit nec nisi vulputate nonummy. Maecenas tincidunt
11		11	Bugün kurs yayınlanmayacaktır!

	announcement_id [PK] integer	instructor_id [PK] integer	message character varying (1000)
1		1	Fusce posuere felis sed lacus. Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue
2		2	Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue, risus semper porta volutpat, c
3		3	In eleifend quam a odio. In hac habitasse platea dictumst. Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut
4		4	Fusce congue, diam id ornare imperdiet, sapien urna pretium nisl, ut volutpat sapien arcu sed augue. Aliquam erat volutpat. In congue. Etiam justo. E
5		5	Curabitur gravida nisi at nibh. In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem. Integer
6		6	Donec ut dolor. Morbi vel lectus in quam fringilla rhoncus. Mauris enim leo, rhoncus sed, vestibulum sit amet, cursus id, turpis. Integer aliquet, mass
7		7	Morbi a ipsum. Integer a nibh. In quis justo. Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet. Maecenas leo odio, condim
8		8	Duis mattis egestas metus. Aenean fermentum. Donec ut mauris eget massa tempor convallis. Nulla neque libero, convallis eget, eleifend luctus, ult
9		9	Vestibulum quam sapien, varius ut, blandit non, interdum in, ante. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Cu
10		10	Ut tellus. Nulla ut erat id mauris vulputate elementum. Nullam varius. Nulla facilisi. Cras non velit nec nisi vulputate nonummy. Maecenas tincidunt l
11		11	Yarın C++ kursu geliyor!

```
void update_row(PGconn *conn, std::string table_name, std::string column, std::string condition, std::string data){  
    std::string query = "update ";  
    query.append(str: table_name);  
    query.append(s: " set ");  
    query.append(str: column);  
    query.append(s: " = ");  
    query.append(str: data);  
    query.append(s: " where ");  
    query.append(str: condition);  
    query.append(s: ";");  
  
    PGresult *res = PQexec(conn, query.c_str());  
    std::cout << PQerrorMessage(conn);  
}
```