

□ جلسه ۲۱ – مقدمه‌ای بر ساخت مینی پروژه با پایگاه داده

در این جلسه، یک پروژه ساده با استفاده از پایگاه داده (Database) شروع می‌شود که در آن مفاهیم پایه‌ای مانند جدول‌ها، کلید خارجی (Foreign Key) و ایندکس (Index) مورد بررسی قرار می‌گیرد.

📦 ساخت جداول پایگاه داده

ابتدا دو جدول اصلی در پایگاه داده ساخته می‌شوند:

1. جدول notes

```
CREATE TABLE notes (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    body TEXT  
) ENGINE=InnoDB;
```

- id: شماره یادداشت (کلید اصلی) PRIMARY KEY -
- body: متن یادداشت

2. جدول users

```
CREATE TABLE users (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    name NVARCHAR(255),  
    email VARCHAR(255)  
) ENGINE=InnoDB;
```

- id: شماره کاربر (کلید اصلی)
 - name: نام کاربر
 - email: ایمیل کاربر
-

🔒 اضافه کردن ایندکس (Index) به ایمیل برای جلوگیری از تکراری بودن

برای اینکه ایمیل هر کاربر یکتا باشد، یک ایندکس یکتا روی فیلد email قرار می‌دهیم:

```
CREATE UNIQUE INDEX email_index ON users (email);
```

🔗 اضافه کردن کلید خارجی به جدول notes

برای ارتباط بین یادداشت‌ها و کاربران، یک فیلد جدید به جدول notes اضافه می‌کنیم:

```
ALTER TABLE notes ADD COLUMN user_id INT UNSIGNED;
```

و سپس این فیلد را به عنوان کلید خارجی تعریف می‌کنیم:

```
ALTER TABLE notes  
ADD CONSTRAINT fk_user_note  
FOREIGN KEY (user_id)  
REFERENCES users(id);
```

نکته مهم: نوع داده user_id باید کاملاً مشابه با users.id باشد (مثلاً هر دو INT UNSIGNED) تا کلید خارجی به درستی ساخته شود.

□ تست کردن ارتباط کلید خارجی

برای تست ارتباط بین جدول‌ها:

۱. یک رکورد در جدول users وارد می‌کنیم.
 ۲. یک یادداشت (note) با user_id همان کاربر وارد می‌کنیم.
 ۳. اگر تلاش کنیم یادداشتی با user_id نامعتبر وارد کنیم یا کاربری را حذف کنیم که یادداشت دارد، خطا خواهیم گرفت.
-

□ حذف یا وارد کردن اطلاعات به جدول notes

در طول آموزش، اطلاعاتی وارد و سپس پاک می‌شوند تا مفهوم کلید خارجی و وابستگی بین داده‌ها به‌صورت عملی توضیح داده شود.

✦ نکات مهم

- هنگام استفاده از کلید خارجی، مطمئن شوید هر دو جدول از موتور InnoDB استفاده می‌کنند.
 - نام فیلدها، نوع داده‌ها و ترتیب ساخت جدول‌ها باید دقیق باشد.
 - کلید خارجی فقط زمانی قابل اضافه‌شدن است که جدول مقصد دارای کلید اصلی (Primary Key) یا ایندکس یکتا (Unique Index) باشد.
-

□ توضیح کامل دستورات SQL استفاده‌شده در جلسه ۲۱

در این بخش، تمام دستورات SQL که در جلسه ۲۱ استفاده شد، به‌صورت کامل و خطبه‌خط توضیح داده می‌شود تا برای افراد تازه‌کار کاملاً قابل‌فهم باشد.

✓ CREATE TABLE – ساخت جدول جدید

```
CREATE TABLE notes (  
    id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    body TEXT  
) ENGINE=InnoDB;
```

این دستور یک جدول به نام notes ایجاد می‌کند:

- `id INT UNSIGNED`: یک عدد صحیح مثبت است که به عنوان شناسه (ID) هر یادداشت استفاده می‌شود.
- `AUTO_INCREMENT`: به‌صورت خودکار به هر رکورد جدید یک شماره یکتا می‌دهد.
- `PRIMARY KEY`: این فیلد را به عنوان کلید اصلی جدول مشخص می‌کند.

- body TEXT: متن یادداشت را ذخیره می‌کند.
 - ENGINE=InnoDB: نوع موتور پایگاه داده. برای استفاده از کلید خارجی، باید InnoDB باشد.
-

✓ CREATE UNIQUE INDEX – ایجاد ایندکس یکتا

```
CREATE UNIQUE INDEX email_index ON users (email);
```

- این دستور روی ستون email در جدول users یک ایندکس ایجاد می‌کند.
- UNIQUE: به این معناست که هیچ دو کاربری نمی‌توانند ایمیل یکسان داشته باشند.
- email_index: نام دلخواهی است که برای ایندکس انتخاب شده (می‌تواند هر اسمی باشد).

✦ ایندکس چیست؟

ایندکس (Index) مثل فهرست کتاب است. کمک می‌کند پایگاه داده سریع‌تر رکوردها را بر اساس یک فیلد پیدا کند. وقتی UNIQUE هم باشد، علاوه بر سرعت، جلوی تکرار داده‌ها را می‌گیرد.

✓ ALTER TABLE – تغییر جدول موجود

برای اضافه کردن یک ستون جدید:

```
ALTER TABLE notes ADD COLUMN user_id INT UNSIGNED;
```

- این دستور به جدول notes یک ستون جدید به نام user_id اضافه می‌کند.
- این فیلد برای مشخص کردن این است که هر یادداشت متعلق به کدام کاربر است.

برای تعریف کلید خارجی:

```
ALTER TABLE notes  
ADD CONSTRAINT fk_user_note  
FOREIGN KEY (user_id)
```

```
REFERENCES users(id);
```

- `ADD CONSTRAINT fk_user_note:` یک محدودیت (constraint) جدید با نام `fk_user_note` تعریف می‌شود.
- `FOREIGN KEY (user_id):` مشخص می‌کند که ستون `user_id` در جدول `notes` به صورت کلید خارجی عمل کند.
- `REFERENCES users(id):` یعنی مقدار این ستون باید با مقدار `id` در جدول `users` مطابقت داشته باشد.

★ کلید خارجی: (Foreign Key)

برای ایجاد ارتباط بین دو جدول استفاده می‌شود. در این مثال، هر یادداشت (note) به یک کاربر (user) متصل می‌شود.

⚠ خطای معروف 150: `errno: 150`

اگر هنگام اضافه کردن کلید خارجی با این خطا روبرو شدید:

```
Can't create table ... (errno: 150 "Foreign key constraint is incorrectly formed")
```

علت‌های رایج:

- نوع ستون‌ها در دو جدول یکی نیست (مثلاً یکی `INT` و دیگری `INT UNSIGNED`).
- جدول‌ها از موتور `InnoDB` استفاده نمی‌کنند.
- جدول `users` هنوز ساخته نشده یا `id` کلید اصلی نیست.
- سعی کردید کلید خارجی روی فیلدی بگذارید که داده‌های ناسازگار دارد.

ساخت جدول notes با فیلدی id(int),body(text)

ساخت جدول users با فیلد id(int),name(nvarchar(255)),email(varchar(255))

بعد کل پایگاه داده خارج شده دوباره وارد میشویم یک index برای فیلد email برای یکتا بودنش درست میکنیم.

بعد یه فیلد user-id برای جدول notes ایجاد کرده و به عنوان کلید خارجی تعریف میشود سپس اطلاعات درونش وارد میکن و پاک میکند تا مفهوم کلید خارجی رو بیان کند