Лабораторийн тайлан №6

Апплекэйшны өгөгдлийн санг үүсгэн хэрэгжүүлэх

МТЭС, МКУТ, Программ хангамж

Г.Батням, 22B1NUM5578

1.Ажлын зорилго

Энэ лабораторийн ажлаар апплекэйшинд ашиглагддаг буюу тухайн апплекэйшнээс гараад ороход өгөгдөл нь өгөгдлийн санд хадгалагдан илүү хурдан ажиллах боломжтой эсэхийг туршин практикт хэрэгжүүлэх.

2.Онолын судалгаа

Мэдээллийн тогтвортой байдал (data persistence) – Android хөгжүүлэлтийн чухал хэсэг юм. Энэ нь хэрэглэгчийн оруулсан өгөгдөл апп хаагдсан ч хадгалагдах, эсвэл интернетээс татсан өгөгдөл дахин татах шаардлагагүйгээр ашиглагдах боломжтой болгодог.

Холбоост өгөгдлийн сан (relational database):

- **Хүснэгтүүд** нь тодорхой мэдээллийг **бүлэглэн** хадгалдаг. (Жишээ нь: **оюутан**, **багш**, **хичээл** гэх мэт)
- **Баганууд** нь тухайн хүснэгтийн **мэдээллийн төрлийг** тодорхойлдог. (Жишээ нь: **ID**, **нэр**, **мэргэжил**, **дүн** гэх мэт)
- **Мөрүүд** нь тухайн хүснэгтийн **бодит өгөгдлийг** агуулдаг. (Жишээ нь: **нэг оюутны мэдээлэл** гэх мэт)

Анхдагч түлхүүр ба гадаад түлхүүр

- **Анхдагч түлхүүр** (*Primary Key*) – хүснэгт дэх мөрүүдийг **нэг нэгнээс ялгах давтагдашгүй утга** бүхий багана. Ихэвчлэн **автоматаар нэмэгддэг ID** багана анхдагч түлхүүр болдог.

- **Гадаад түлхүүр** (*Foreign Key*) – нэг хүснэгтээс нөгөө хүснэгтийн **анхдагч түлхүүрийг** заах утга. Энэ нь хүснэгтүүдийн хооронд **харилцаа** (relationship) үүсгэхэд хэрэглэгддэг.

Схем – Өгөгдлийн сангийн **бүх хүснэгтүүд** болон **баганууд**ын бүтэц нь **схем** (*schema*) гэж нэрлэгддэг.

- COUNT(): Тухайн хүснэгтэнд байгаа нийт мөрийн тоог буцаана.
- SUM(): Тухайн хүснэгтээс тухайн элемэнтийн нийлбэрийг буцаана.
- AVG(): Тухайн хүснэгтээс тухайн элемэнтийн дунджийг буцаана.
- MIN(): Тухайн хүснэгтээс тухайн элемэнтийн хамгийн бага утгыг буцаана.
- MAX(): Тухайн хүснэгтээс тухайн элемэнтийн хамгийн их утгыг буцаана.
- DISTINCT(): Тухайн хүснэгтээс тухайн элемэнтийн давхацлыг хасан буцаана.
- WHERE(): Тухайн хүснэгтээс тухайн элемэнтийг тодорхой нөхцөлтэйгөөр харах боломжтой.
- LIKE():
  1. %search term% - Бүх утга дотор тодорхой текст агуулсан өгөгдлийг хайх
  2. %search term - Тухайн текстээр эхэлсэн утгуудыг хайх
  3. search term% - Тухайн текстээр төгссөн утгуудыг хайх
- GROUP BY(): Нийт элемэнтийн утгыг давхацаагүй утгаар бүлэглэх боломжтой
- ORDER BY(): Тухайн нэг мөрийн утгаар нь өсөхөөр болон буурахаар эрэмбэлэх боломжийг олгодог.
- LIMIT(): Тухайн хүснэгтээс нийт хэдэн мөрийг авах хязгаарыг тодорхойлсон
- LIMIT OFFSET(): Энэ нь эхний 10-ыг алгасаад, 11-20 дахь имэйлүүдийг авна. **LIMIT 10 OFFSET 10;**

Reactive –

StateFlow – Kotlin Flow-ийн нэг төрөл бөгөөд реактив архитектурт төлөв байдлыг удирдах зориулалттай. Энэ нь төлөв байдлыг хадгалж, шинэчлэх, мөн төлөв байдалд хийх өөрчлөлтүүдийг ажиглах боломжийг олгодог. Энгийн **Flow**-оос ялгаатай нь, **StateFlow** нь дангаараа **State** буюу тухайн обьектын төлөв байдлыг хадгалж байдаг.

StateFlow-ийг ашиглах давуу талууд:

- **Хялбар удирдлага:** Төлөв байдлыг нэг газарт хадгалж, ашиглахад амархан.
- **Реактив:** Төлөв байдалд ямар нэгэн өөрчлөлт гарвал түүнийг хүлээн авч, шууд ашиглаж болно.
- **Concurrency (хэрэглэгчийн олон үйлдэл):** Төлөв байдал нь thread-safe бөгөөд хэд хэдэн үүрэг гүйцэтгэгчтэй орчинд зөв зохион байгуулагддаг.

StateFlow – Хэзээ ч **хоосон** байхгүй. Тэр нь үргэлж нэг утгыг хадгалдаг, мөн энэ утгыг та **үнэмлэхгүй** үед ч гэсэн үргэлж нэг утгыг өгч байдаг. Үндсэндээ, **StateFlow** нь төлөв байдлыг хадгалж, хамгийн сүүлийн утга нь үргэлж байх болно. **StateFlow** нь зөвхөн утга өөрчлөгдөхөд хэрэглэгчдэд мэдээлдэг. Та **StateFlow**-ийн утгыг өөрчилснөөр бүх ажиглагчид (collect) тэр шинэ утгыг хүлээн авна.Тухайн layer хоорондын төлөвтөө нэг утга өгөөд түүнийгээ хэрэгжүүлээд өөрчлөөд шинэчлэх зарчмаар явагддаг.Ui layer болон data layer хооронд төлөв хадгалах үйлдлийг хийж гүйцэтгэдэг.Тийм учраас бид

Flow – **Төлөв байдлыг хадгалахгүй**. Энэ нь үргэлж өгөгдлийн урсгалыг дамжуулдаг бөгөөд та **үргэлж шинэ өгөгдөл хүлээн авах** болно. **Flow**-д одоо байгаа утга байхгүй, зөвхөн асинхрон өгөгдлийн урсгал дамжуулагддаг.Хуучин өгөгдлийг хадгалдаггүй учир нь flow нь урсгал юм.Ui layer болон data layer хоёрын хооронд тухайн өгөгдлийг дамжуулах зорилготой.Тухайн утгыг хадгалахгүй дамжуулах ажлыг хийж гүйцэтгэдэг.

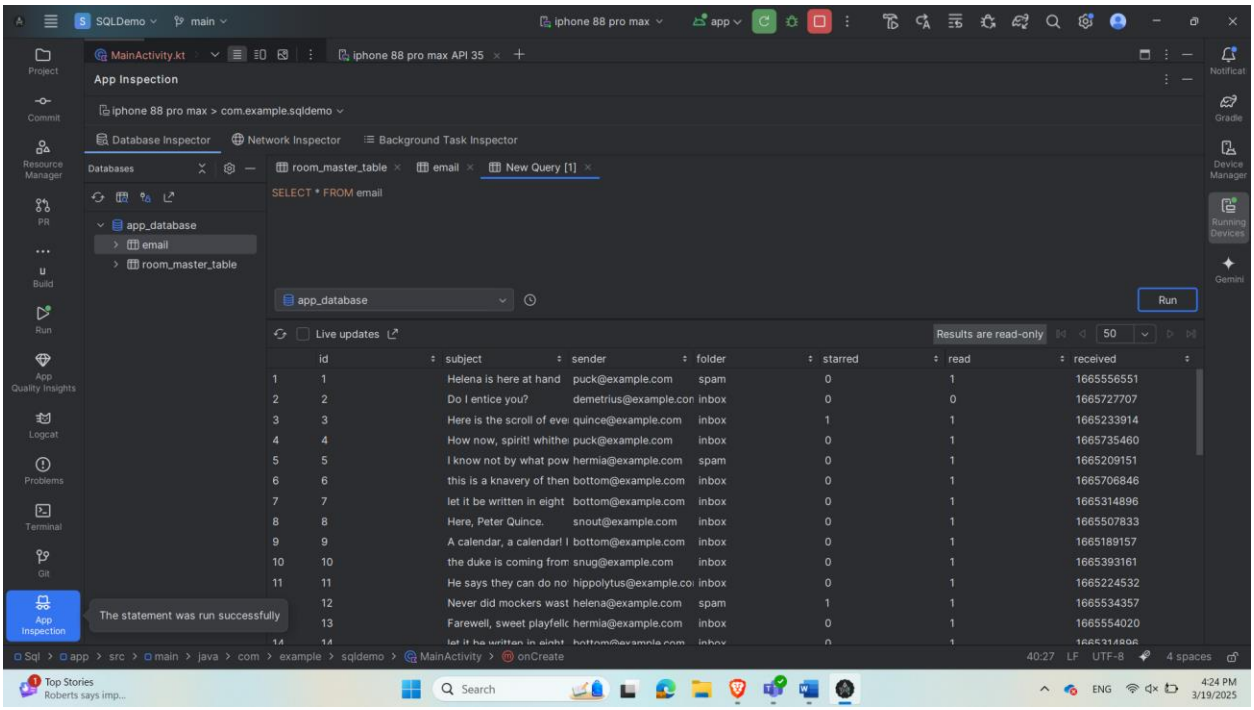
@ColumnInfo – Ингэж зарласнаар тухайн баганын төрлийг өөрчлөх боломжтой болдог.

Room – **Android Jetpack**-ийн өгөгдөл хадгалах **(persistence)** сан бөгөөд энэ нь **SQLite** өгөгдлийн сангийн дээрх **абстракц давхарга** болж ажилладаг. Өгөгдлийн сангийн тохиргоо, харилцан үйлчлэлийг хялбарчилдаг бөгөөд **SQL командуудыг бичих үед алдааг шалгах** боломжийг олгодог.

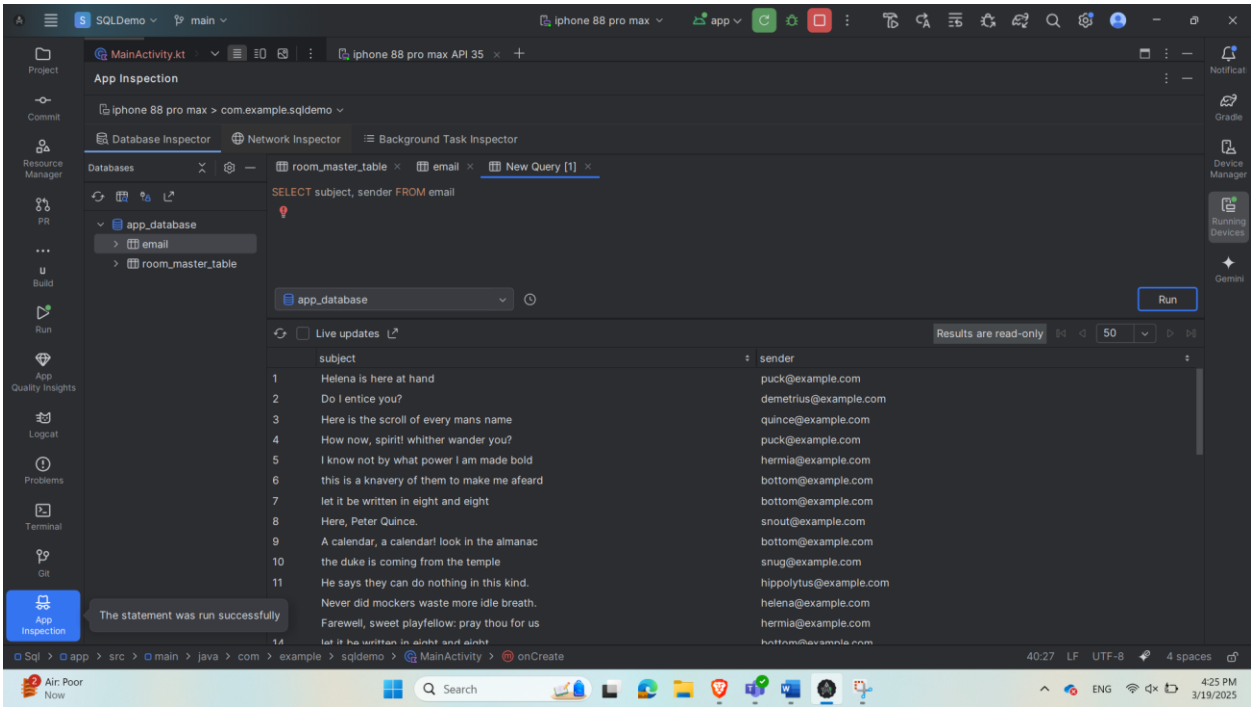
Абстракц давхарга (Abstraction layer) – **Суурь хэрэгжилтийг нууж**, зөвхөн шаардлагатай функцуудыг харуулдаг давхарга юм. Энэ тохиолдолд **Room** нь **SQLite** өгөгдлийн сантай харилцах интерфэйсийг бий болгодог.

NavHostController ашиглан дэлгэц хооронд шилждэг.

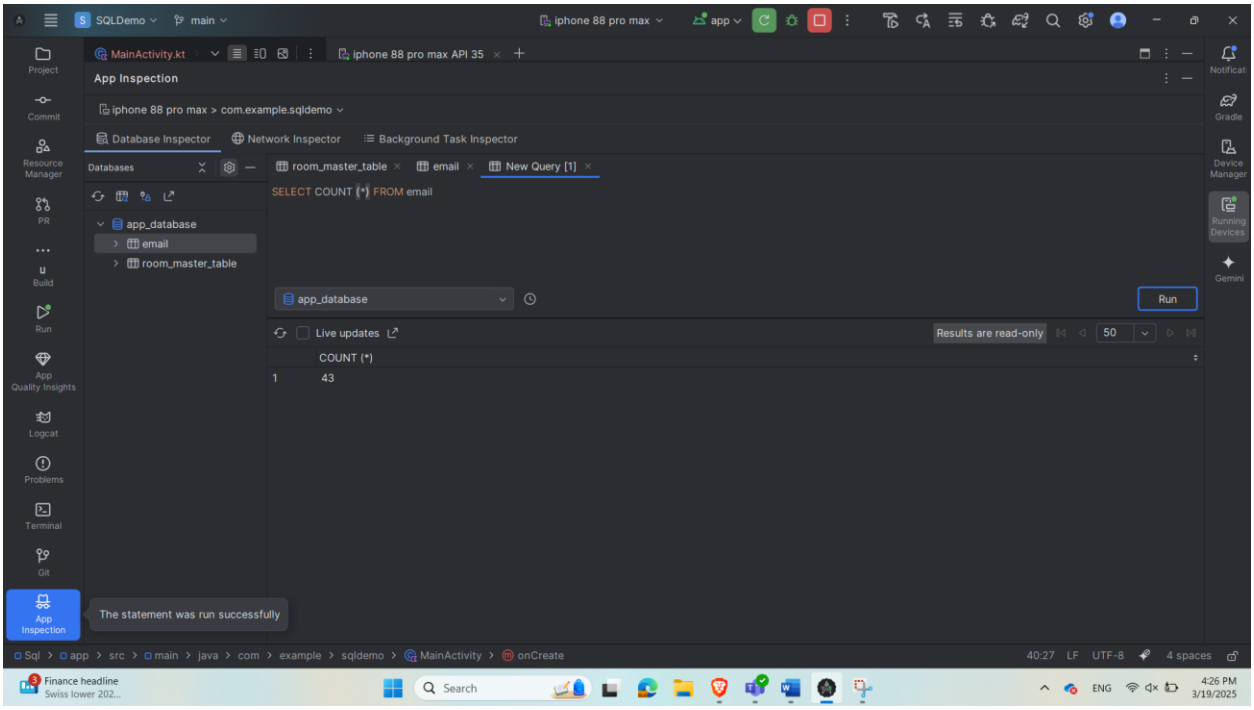
3.Хэрэгжүүлэлт



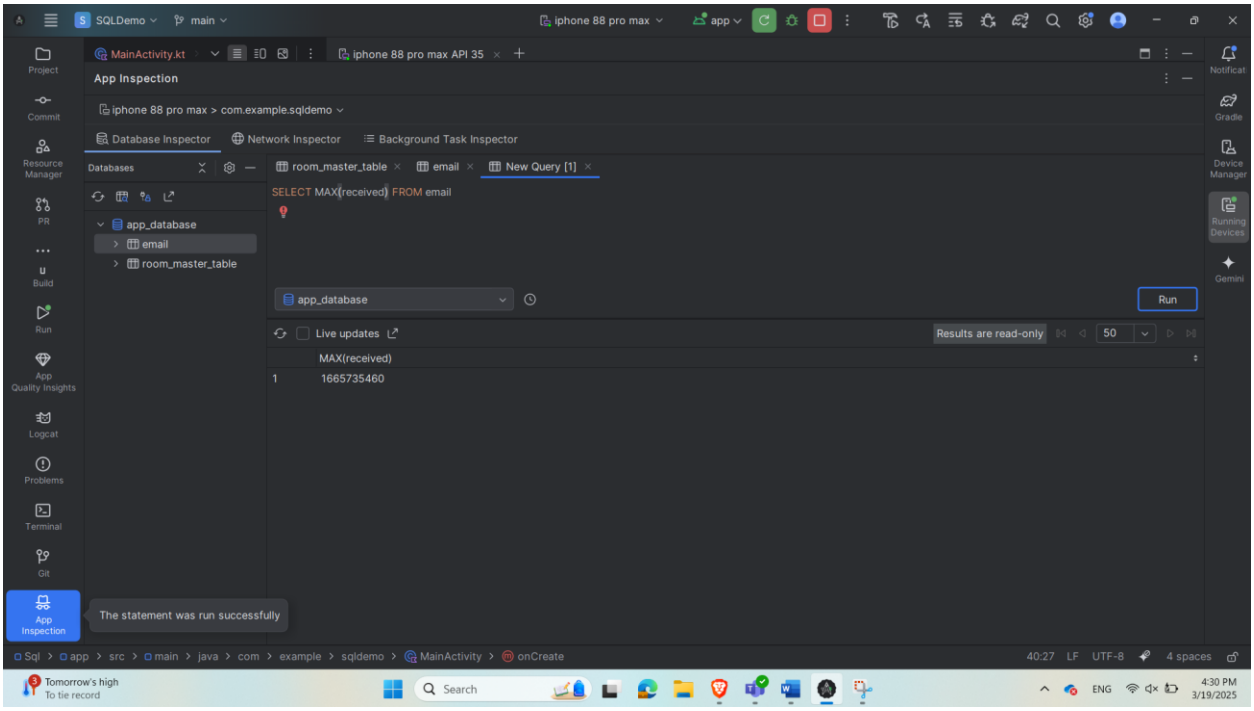
Зураг 1 *Эхний асуулга амжилттай хэрэгжсэн*



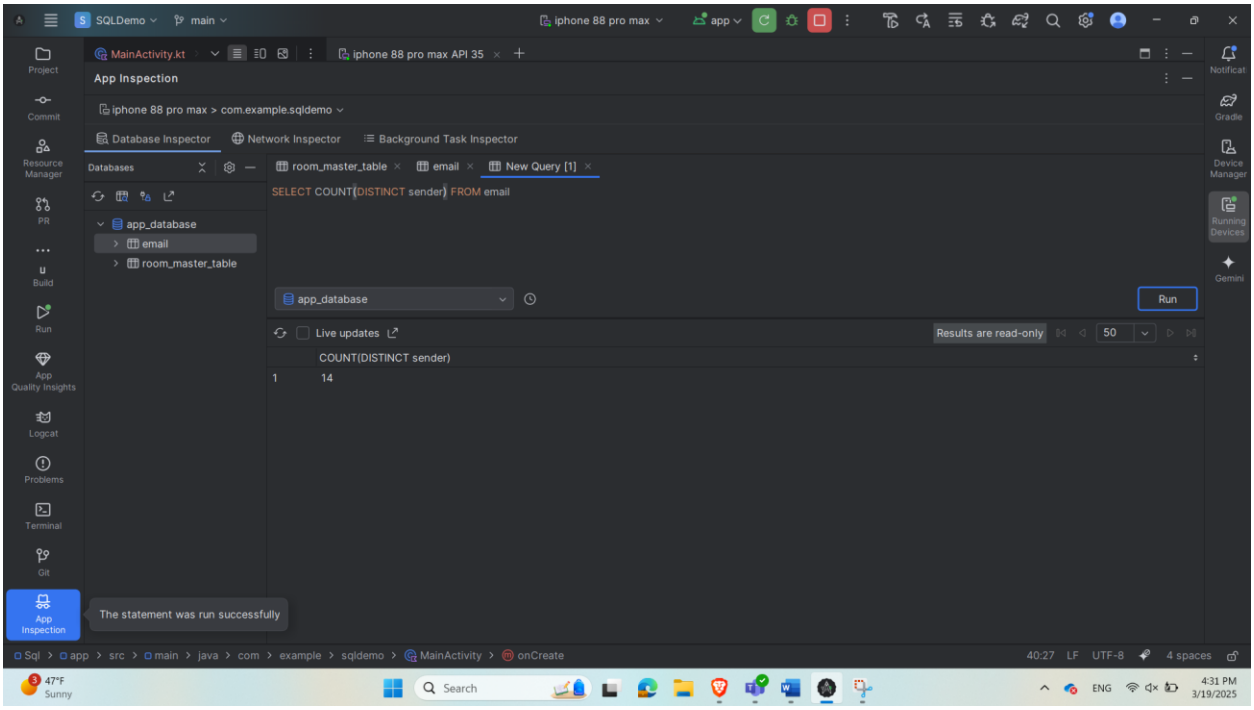
Зураг 2 *2 дах асуулгыг амжилттай хэрэгжүүлсэн*



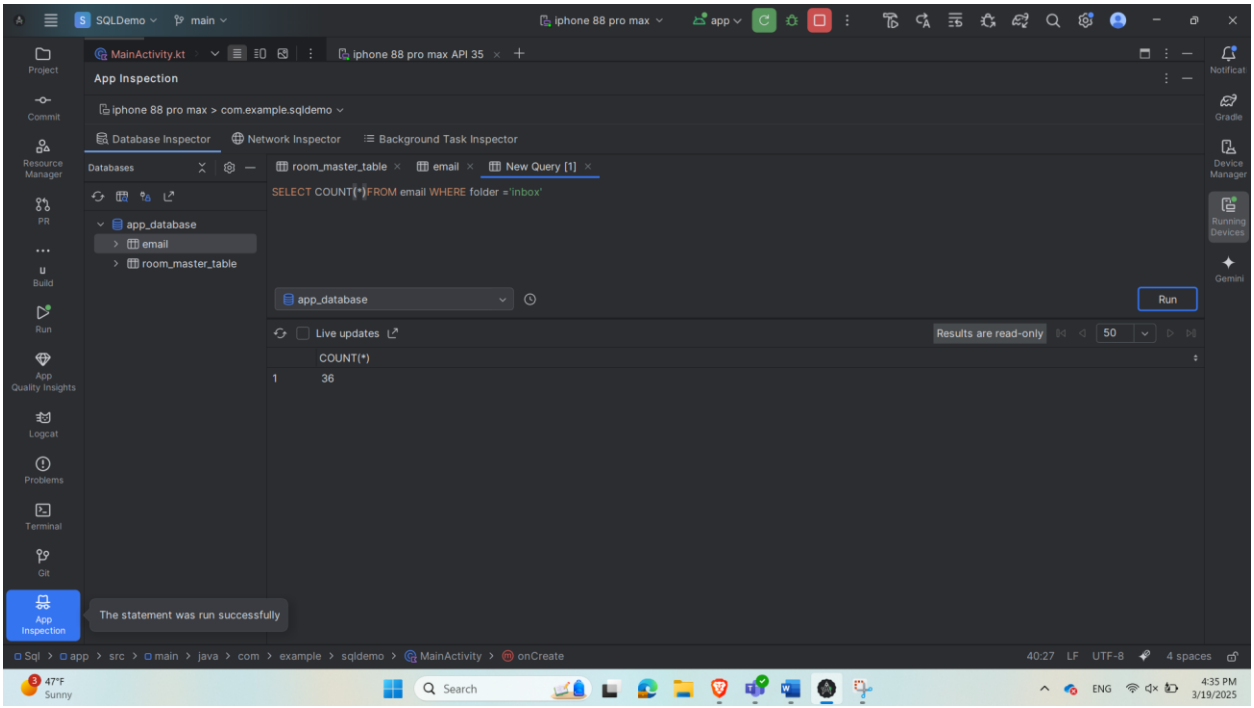
Зураг 3 *Email хүснэгтэнд хэдэн мөртэй мөрийн тоог хэвлэн гаргасан*



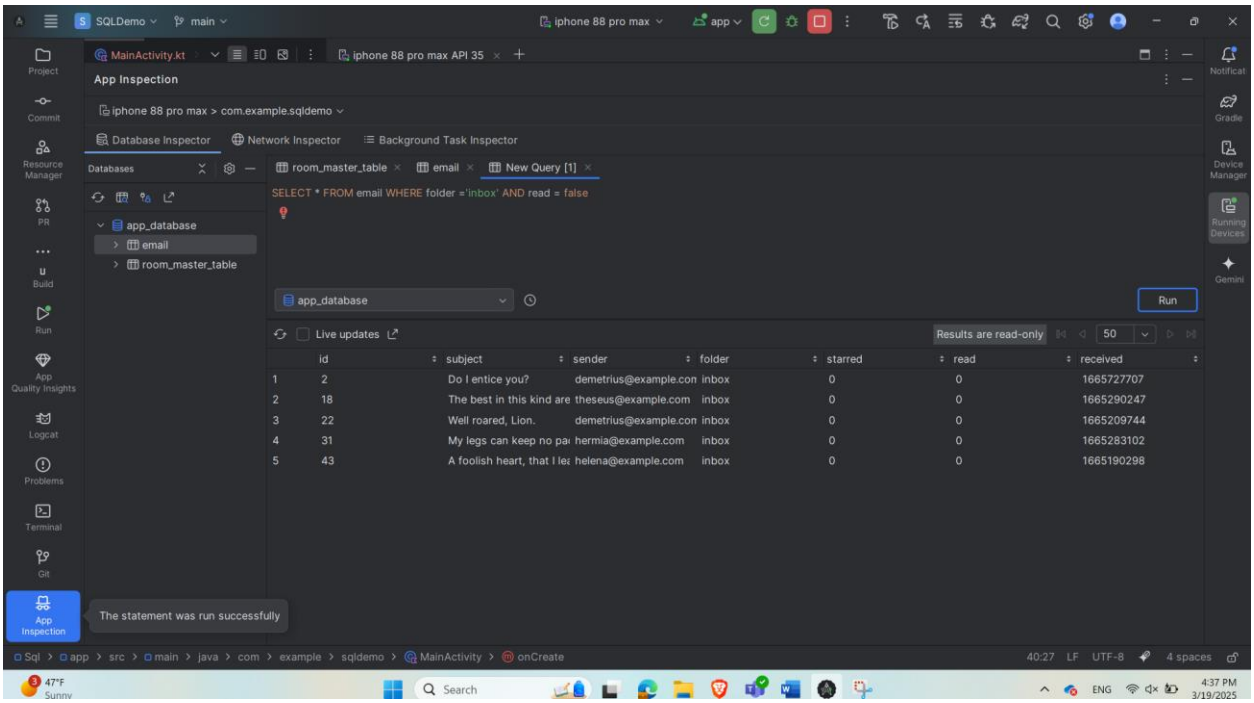
Зураг 4 *Тухайн мөрийн хамгийн их утгыг буцаасан*



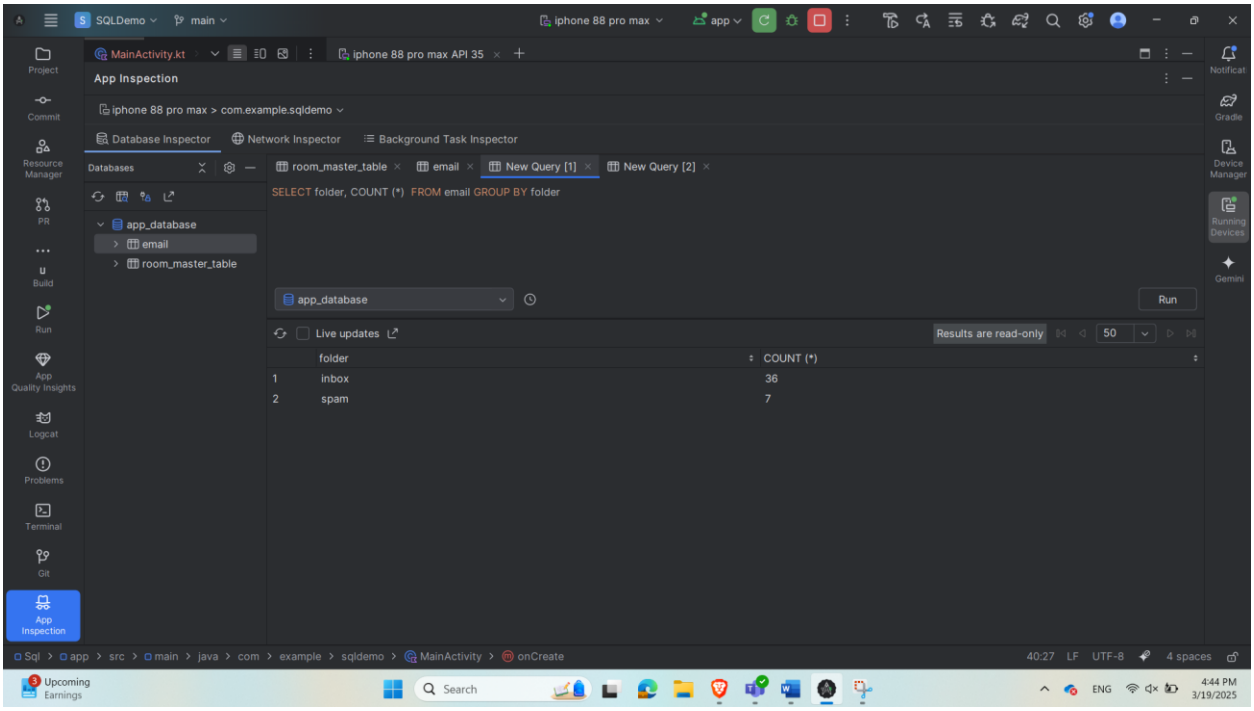
Зураг 5 *sender мөрт давхцаагүй нийт өгөгдлийг харуулсан*



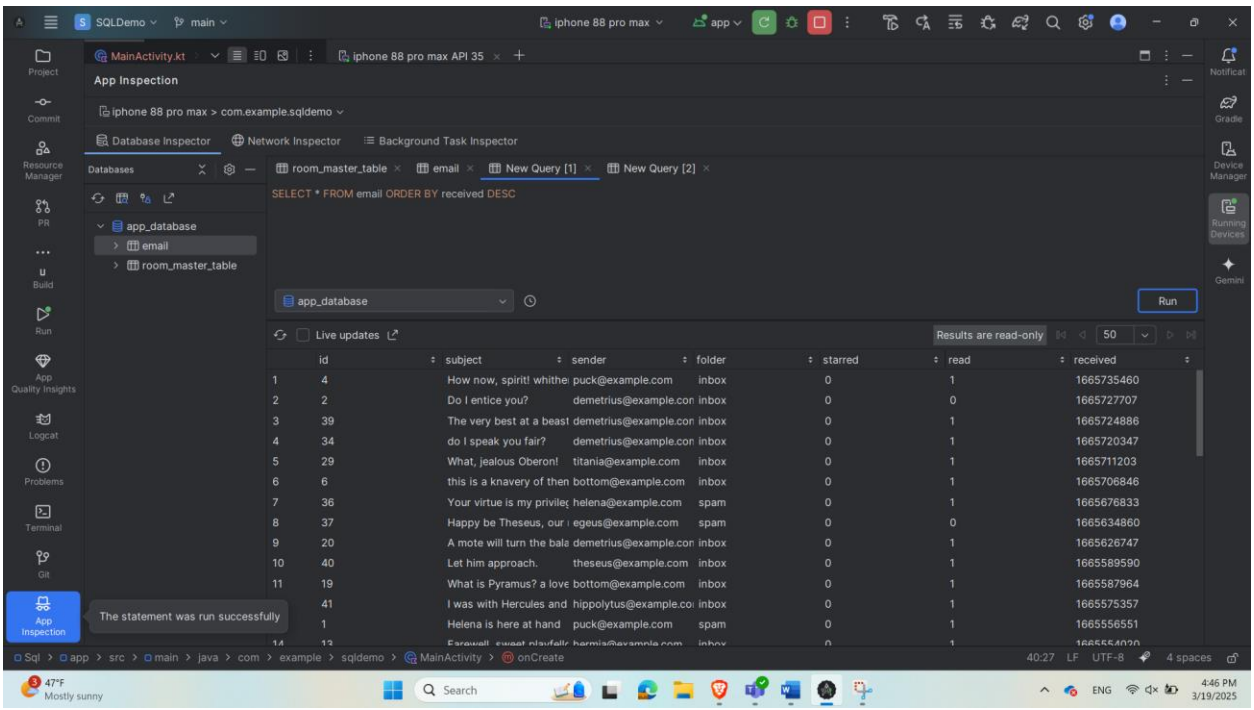
Зураг 6 *Тухайн нийт хүснэгтээс folder нь inbox той тэнцүү байх*



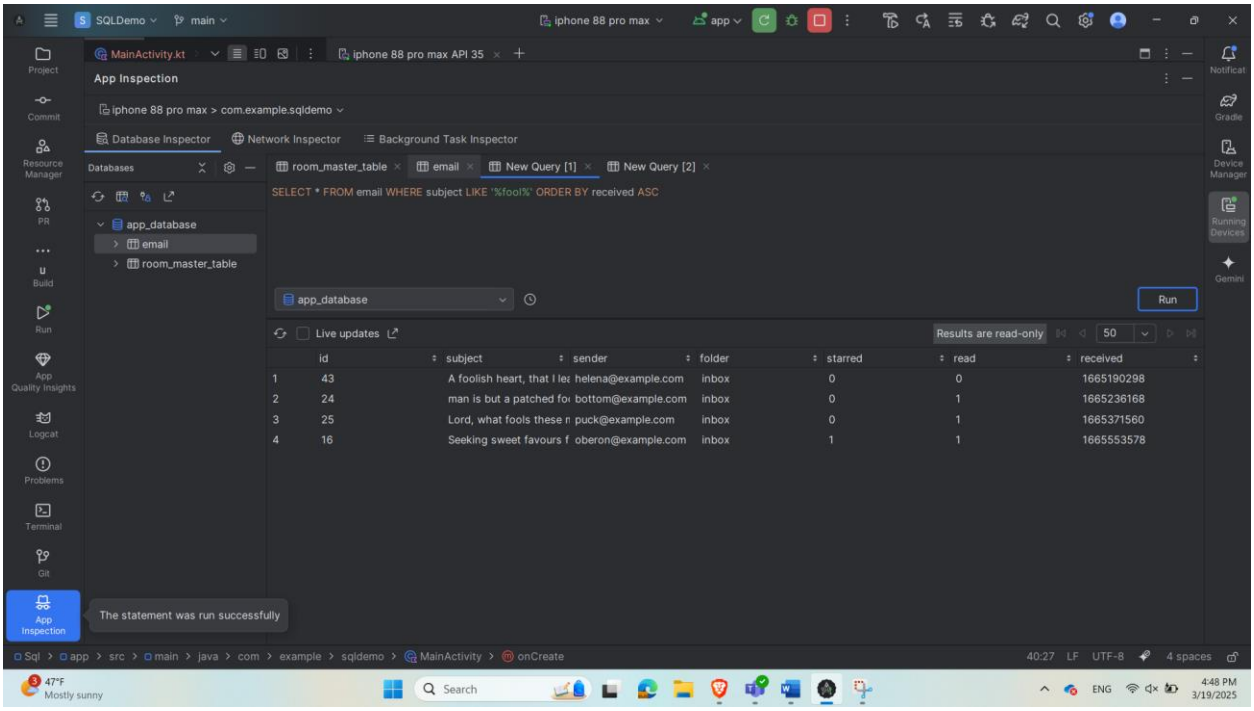
Зураг 7 *AND логикын тусламжтайгаар 2 нөхцөл хангах өгөгдөл харсан*



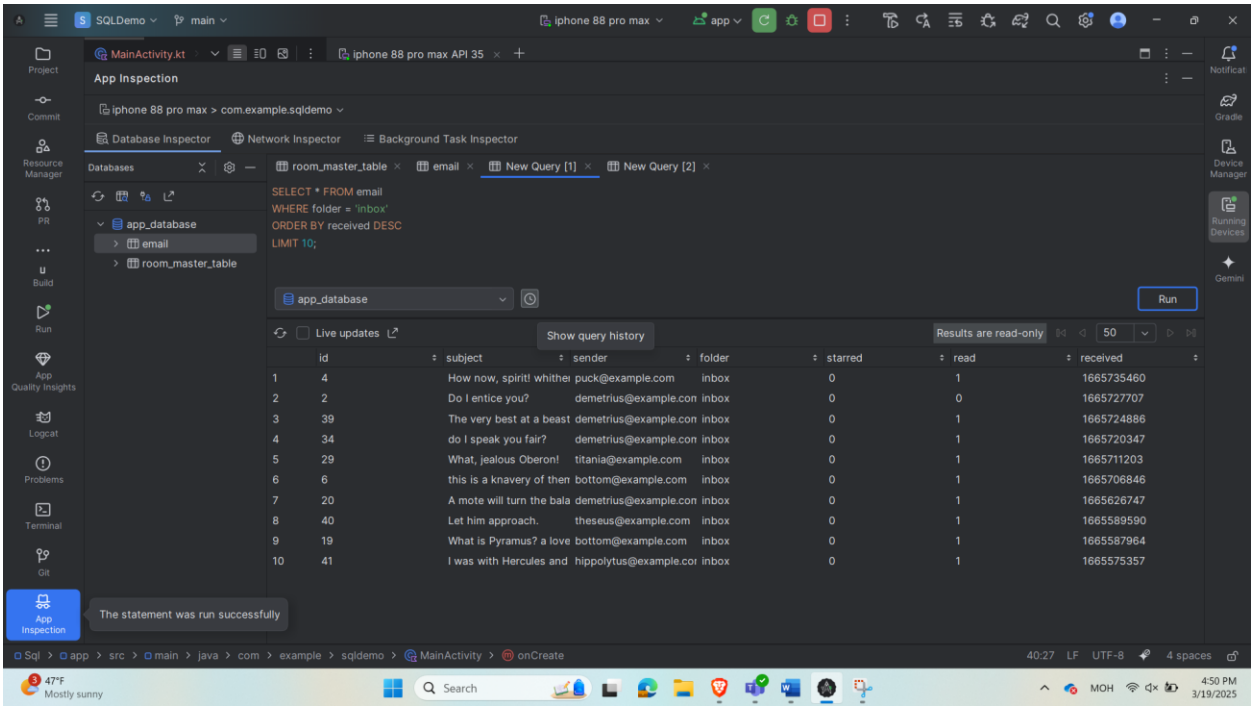
Зураг 8 *folder мөрийн утгуудыг ялгаатай байдлаар нь тоолон бүлэглэсэн*



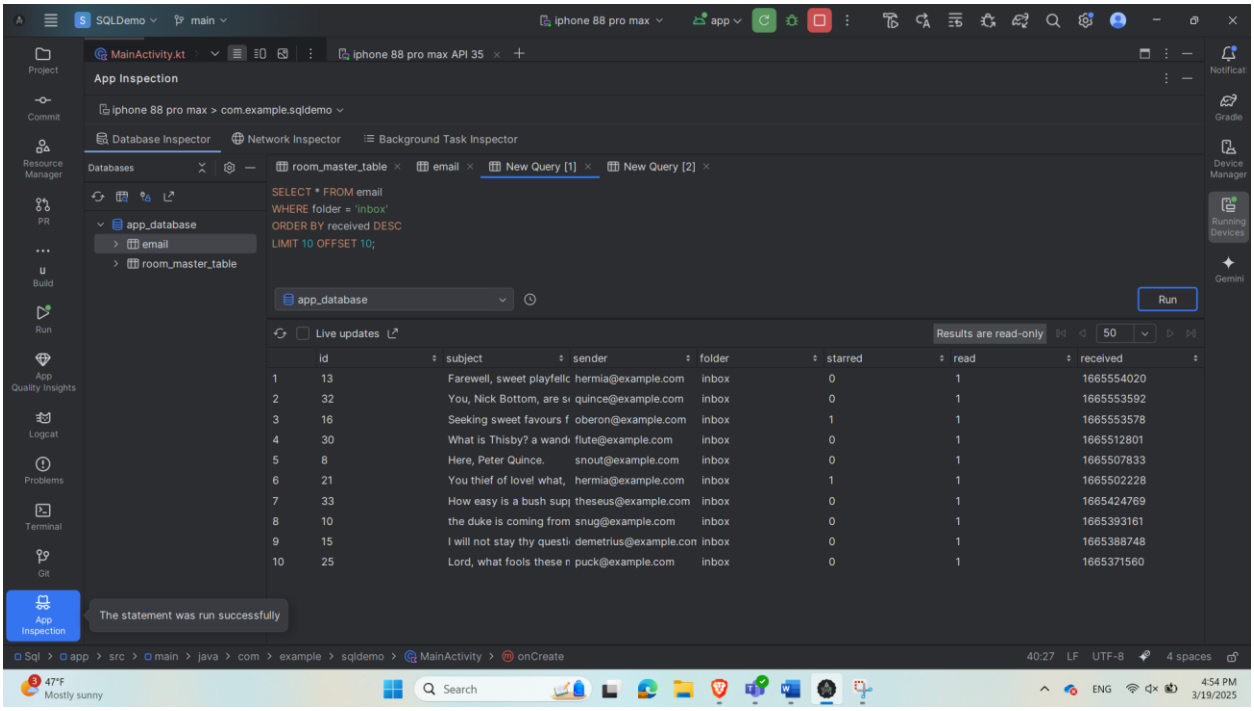
Зураг 9 *Буурахаар эрэмбэлсэн*



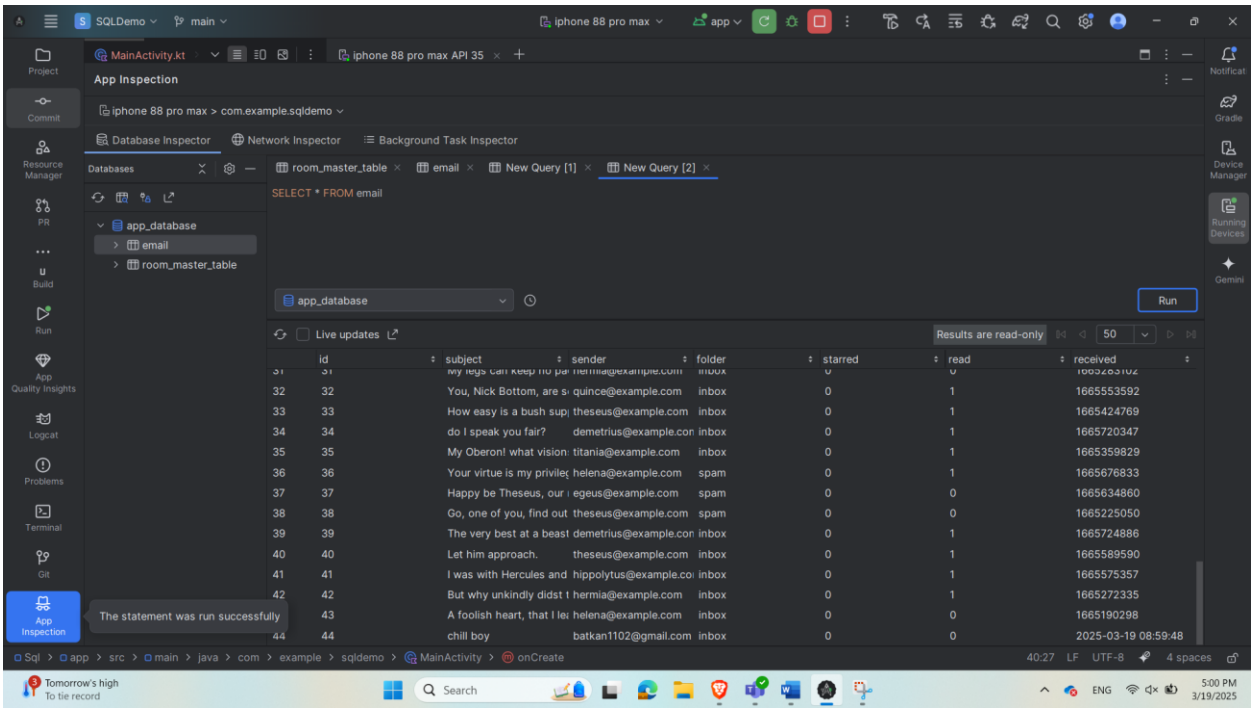
Зураг 10 *WHERE нөхцлийг хангах нийт өгөгдлийг өсөхөөр эрэмбэлсэн*



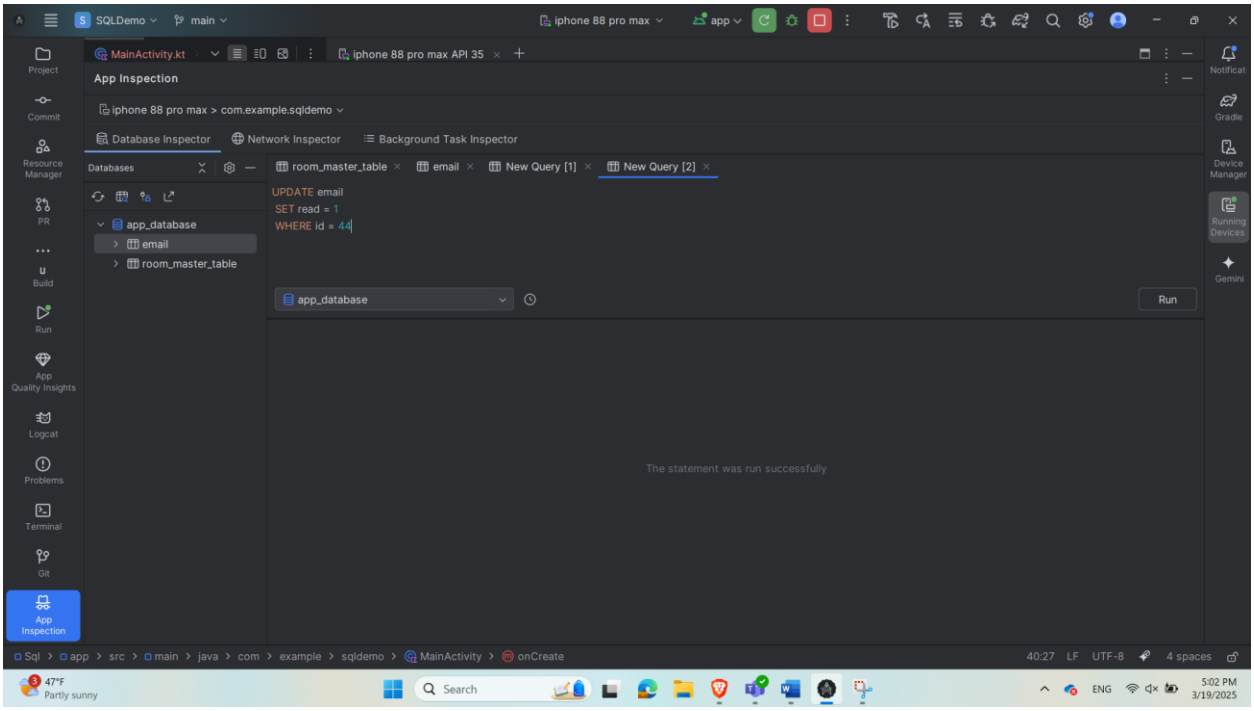
Зураг 11 *Эрэмбэлсэн утгаас хамгийн эхний 10 мөрийг авсан*



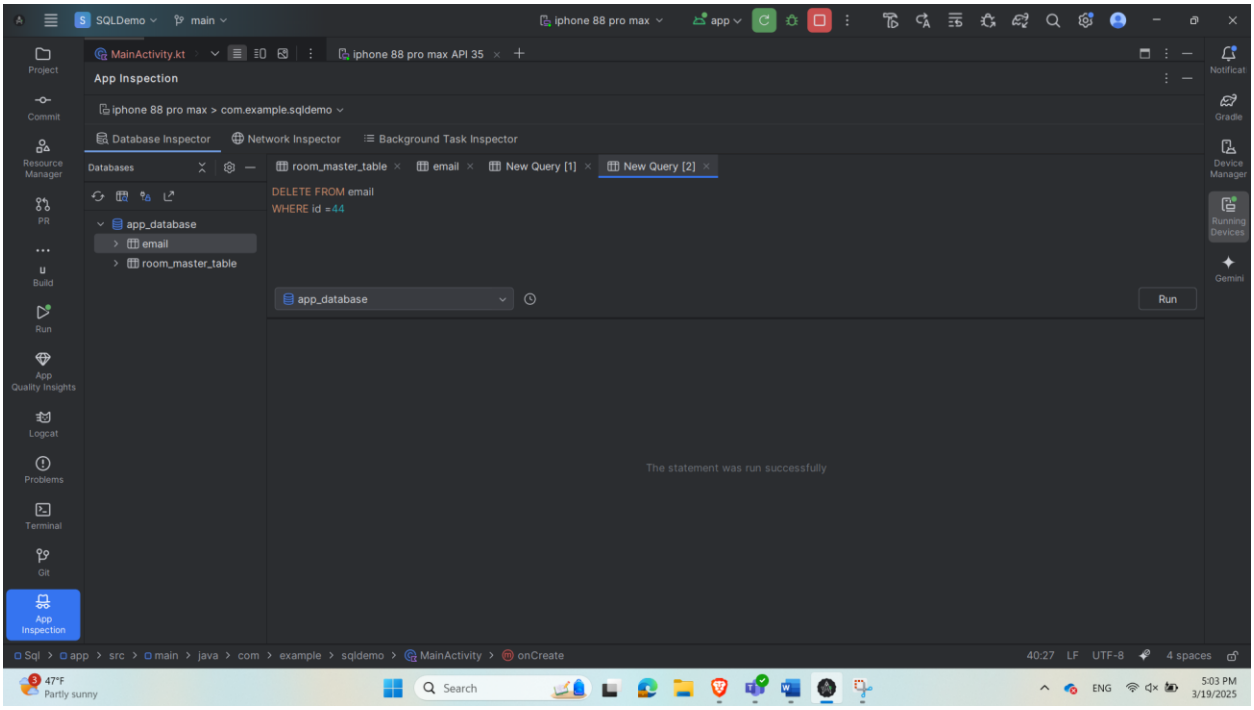
Зураг 12 *Эхний 10 элемэнтийг алгасаад дараагийн 11- с цааших элемэнтээс өгөгдлийг авсан*



Зураг 13 *Шинээр өгөгдөл нэмсэн*

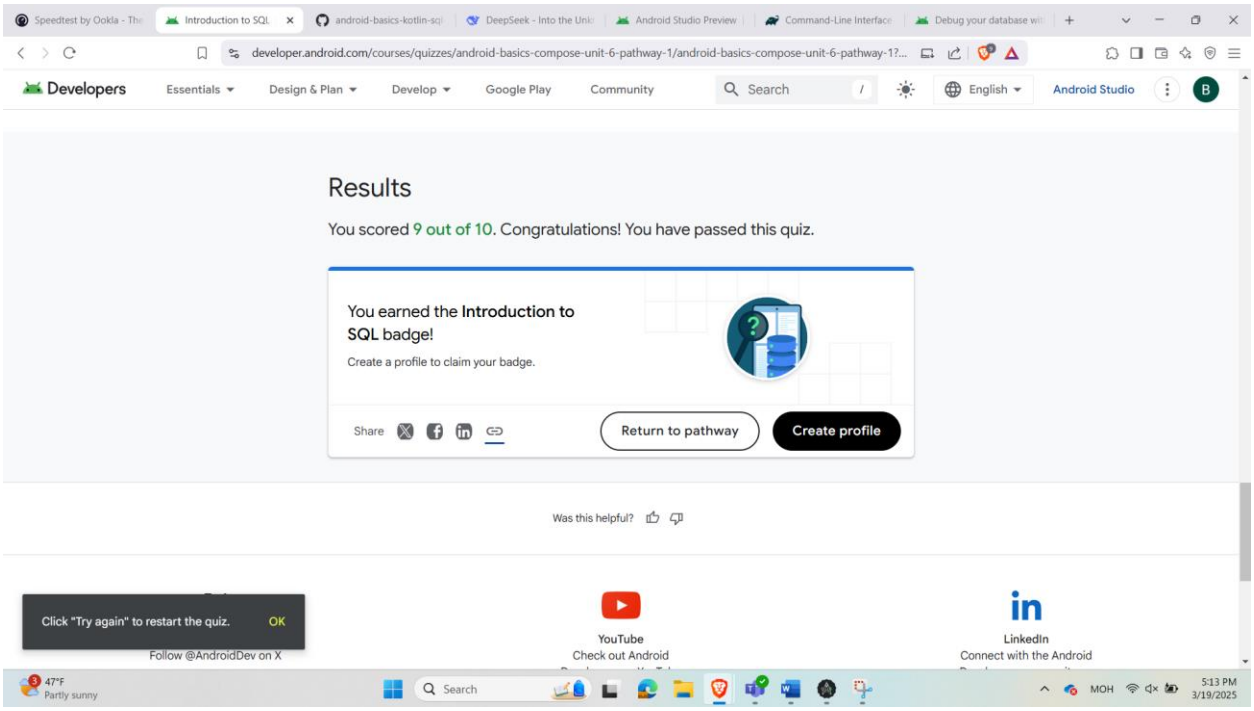


Зураг 14 *Тухайн хүснэгтийн нэг өгөгдөлд өөрчлөлт оруулсан*



Зураг 15 *Тухайн хүснэгтээс нэг өгөгдлийг устгасан*

4. Бүлгийн асуулт



Зураг 1 *Эхний бүлгийн асуултуудад хариулсан*

5.Дүгнэлт

Энэ лабораторийн ажлаар апплекэйшний өгөгдлийн сан үүсгэн өгөгдөл өөрчлөлтийн үр дүнд тухайн урсгалыг ямар flow ашиглан илүү зөв зохицуулж хэрэгцээт мэдээлэл хадгалахыг хийж хэрэгжүүллээ.

6. Ашигласан материал

(https://developer.android.com/courses/pathways/android-basics-compose-unit-3-pathway-1, n.d.)

(https://developer.android.com/courses/pathways/android-basics-compose-unit-3-pathway-2?_gl=1*4nx32f*_up*MQ.., n.d.)

(https://developer.android.com/courses/pathways/android-basics-compose-unit-3-pathway-3, n.d.)