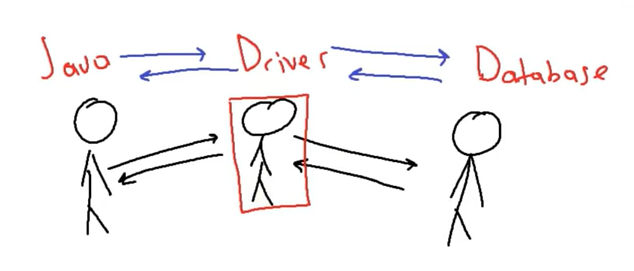
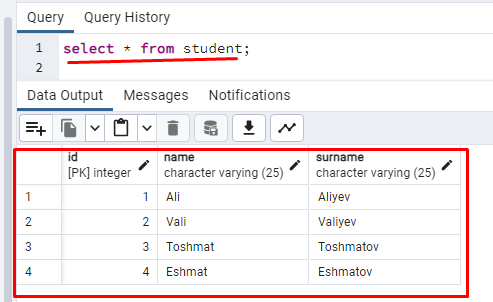
Driver xuddiki tarjimon. Ya’ni Java va Database o’rtasidagi tarjimon bo’lib, Java to’g’ridan-to’g’ri Databaseni tushunmaydi, xuddi shuningdek Database ham Java ni tushunmaydi. Bunday holatda bizga driver yordam beradi. Masalan, MySql uchun o’zini driver mavjud, Oracle uchun alohida va PosgreSql uchun alohida driver bor.



Javada JDBC bilan ishlayotgancha **executeQuery()** methodini parametriga faqat va faqat select berish kerak:



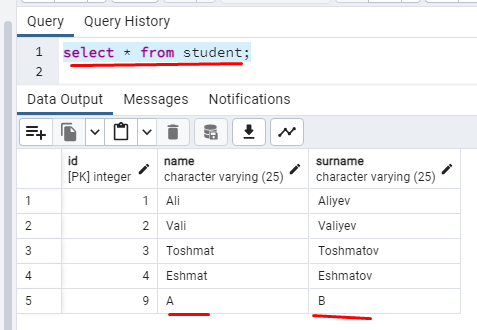
Aks holda agar select ni o’rniga **insert**, **update**, **delete** larni bersak, u holda xatolik beradi. Keling misol ko’raylik. bizda student nomli table bor va 4 ta row mavjud.



Endi shu **executeQuery()** methodiga **insert** commandasini yozadigan bo’lsak, u holda xatolik beradi:

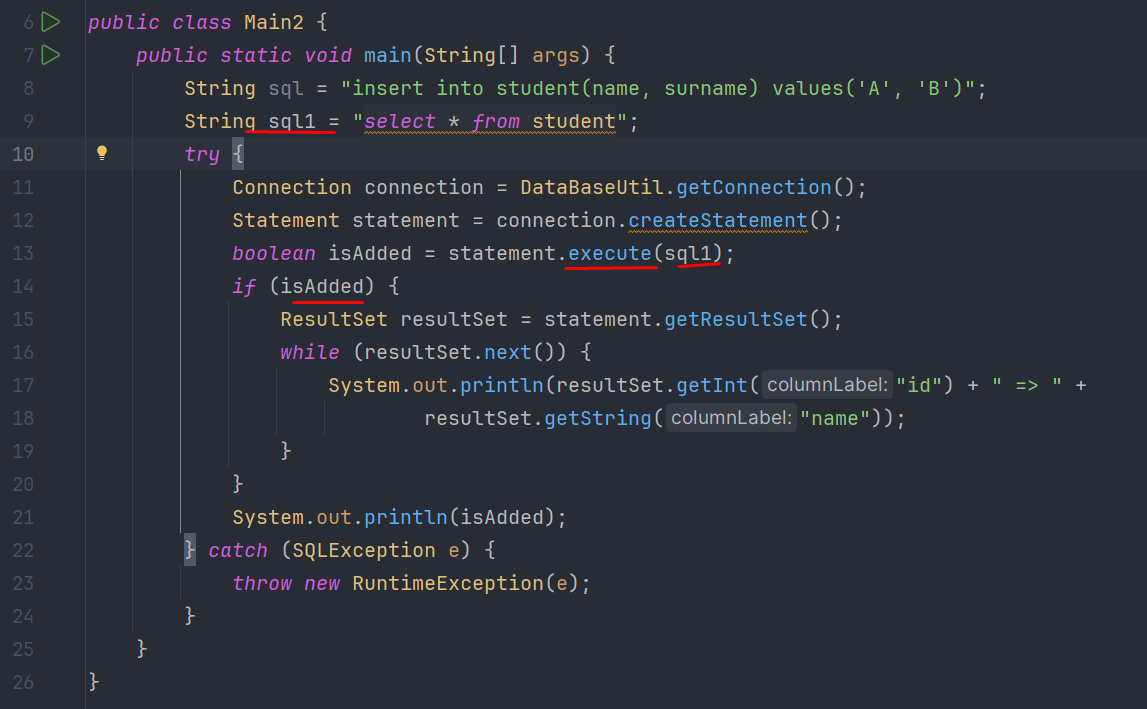


Lekin xato bo’lishiga qaramay, student table mizga “A” va “B” datalarni qo’shadi:



Xo’sh nega qo’shdi? O’zi nima bo’ldi aslida? **executeQuery()** methodi faqat select qabul qiladi. Select ni bilamizki faqat ma’lumot olish uchun ishlatiladi. **executeQuery()** methodi ishga tushganda bizga insert imizdan ma’lumot kelmaydi, to’g’ri nechta row qo’shsak, o’sha qo’shilgan rowlarni sonini qaytaradi. Bu rowlarni sonini Resultset classga o’zlashtirib bo’lmaydi, biz select dan ma’lumot kutyapmiz. Nega xatolik bersa ham tablega qo’shdi, sababi bizni insert imiz bazaga yuboriladi va bazada bajariladi. Natijada bu row bazaga qo’shiladi.

Bundan tashqari select va insert, delete,update larni bittada execute() nomli methodga bersak ham bo’ladi. Xo’sh bu method qanday ishlaydi? Bu method boolean qaytaradi. Agar true qaytsa, demak bizda ResultSet mavjud bu degani executeQuery() methoddan select qilganimzida data qaytadiku, xuddi shunday datalar kelganini bildiradi. Agar false qaytarsa, u holda bizda ResultSet mavjud emas, demak biz select qilmadik balki delete, insert, update, drop qildik deganini bildiradi:



Biz JDBC bilan ishlayotganimizda **Statement** interfacedan tashqari yana **PreparedStatement** degan interface ni ham ishlatishimiz mumkin. Bu ikkisi orasida unchalik katta farq yo’q. Asosiy farqi Statement da SQLInjectiondan himoyalanish ancha pastroq PreparedStatement da esa ancha yaxshiroqdir. Statement da biz insert, update kabi komandalarda parameter sifatida string olganda, shu stringni ichiga sql kodini jo’natib yuborib ishlatishimiz mumkin. PreparedStatement da esa bu narsa anchagina yaxshi himoyalanganda. Bundan tashqari agar sql so’rovimiz parameter sifatida faqat numberni oladigan bo’lsa, u holda Statement ni ham ishlatishimiz mumkin. Chunki number type ga sql kodini yozib bo’lmaydi.