

## Adding Prescription Stored Procedure

The screenshot shows a SQL IDE interface. The top toolbar includes buttons for SQL, Commit, Rollback, and Auto. The main editor displays a stored procedure named 'Add\_Prescription' with the following SQL code:

```
120 --Pickup Person-----
121 --this is needed, bc all prescription will initially have no pickup date
122 --only when they are being picked up, their pickup_id will be updated and their pickup_date change to an actual date
123 insert into pickup_person(pickup_id,
124                             pickup_date)
125 values
126     (nextval('pickup_person_seq'), null);
127
```

Below the editor, there are two tabs: 'patient 1' and 'pickup\_person 2'. The 'pickup\_person 2' tab is active, showing a table with the following data:

	123 pickup_id	pickup_date
1	1	[NULL]

Before executing the Add\_Prescription() procedure, a default pickup\_id is inserted into the pickup\_person table. This pickup is to indicate that the prescription is not picked up, because the pickup date will be null.

The screenshot displays a SQL IDE interface with a dark theme. The top toolbar includes buttons for SQL, Commit, Rollback, and Auto. The main editor shows a stored procedure named `Add_Prescription` with the following code:

```

190 create or replace procedure Add_Prescription(
191     prescribe_drug varchar,
192     patient_lastname varchar, Patient_firstname varchar, patientdob date,
193     prescriber_lastname varchar, prescriber_firstname varchar,
194     pickup_id decimal,
195     serial_number varchar,
196     written_on date,
197     quantity_dispense decimal,
198     date_billed date)
199 as
200 $$
201 declare drug_ndc varchar(11);
202 declare pharm_npi decimal(10);
203 declare get_patient_id decimal(12);
204 declare get_prescriber_npi decimal(10);
205 begin
206     drug_ndc := (select ndc from drug where drug_name = prescribe_drug);
207     pharm_npi := (select pharmacy_npi from pharmacy);
208     get_patient_id := (select patient.patient_id from patient where patient.patient_last_name = patient_lastname
209                       and patient.patient_first_name = patient_firstname
210                       and patient.patient_dob = patientdob);
211     get_prescriber_npi := (select prescriber.prescriber_npi from prescriber where prescriber.Prescriber_last_name = prescriber_lastname
212                             and prescriber.prescriber_first_name = prescriber_firstname);
213
214     insert into prescription(
215         Prescription_number,
216         ndc,
217         pharmacy_npi,
218         patient_id,
219         prescriber_npi,
220         pickup_id,
221         serial_number,
222         written_on,
223         quantity_dispense,
224         date_billed)
225     values (nextval('Prescription_seq'),
226         drug_ndc,
227         pharm_npi,
228         get_patient_id,
229         get_prescriber_npi,
230         pickup_id,
231         serial_number,
232         written_on,
233         quantity_dispense,
234         date_billed);
235
236     update inventory
237     set tobe_filled_quantity = (select tobe_filled_quantity from inventory where ndc = drug_ndc) + quantity_dispense
238     where ndc = drug_ndc;

```

Below the editor, the 'Statistics 1' tab is active, showing a table with two columns: 'Name' and 'Value'. The table is currently empty. The bottom status bar indicates 'Undated Rows: 0', 'Save', 'Cancel', 'Script', and other execution controls. It also shows '200' rows, '1' row(s) updated, and a timestamp of '2022-10-17 at 20:48:14'.

This stored procedure is for adding prescription data into the prescription table. The stored procedure takes the drug's name, patient's information (last name, first name, and date of birth), prescriber's first and last name, pickup\_id, prescription serial number, written\_on date, drug quantity, and the date that it was billed.

To make the query shorter, there are a few variables that are declared in this query. Variable `drug_ndc` returns the drug ndc from the drug table given the drug name argument. Variable `get_patient_id` returns the patient\_id from the patient table given the arguments last name, first name, and date of birth. Variable `get_prescriber_npi` returns the prescriber npi from the prescriber table given the argument prescriber last name and first name.

The stored procedure inserts `prescription_number`, `ndc`, `pharmacy_npi`, `patient_id`, `prescriber_npi`, `pickup_id`, `serial_number`, `written_on`, `quantity_dispense`, and `date_billed` into the prescription table. The `prescription_number` will be from the 'prescription\_seq' sequence. `Ndc`, `patient_id`, and `prescriber_id` are all from the variables declared earlier. `Pickup_id`,

serial\_number, written\_on, quantity\_dispende, and date\_billed are all from the respective arguments.

The stored procedure also updates the inventory and sets the tobe\_filled\_quantity to the sum of the previous tobe\_filled\_quantity with the current prescription's quantity\_dispende argument given the corresponding ndc. This is to track that an added prescription is to be filled.

The screenshot displays a SQL IDE interface with a dark theme. The top toolbar includes icons for SQL, Commit, Rollback, and Auto. The main editor shows a stored procedure named 'add\_prescription' with the following code:

```
186 --Prescription population
187 --arguments: drug_name, pt_lastname, pt_firstname, pt_dob, prescriber_lastname, prescriber_firstname, pickup_id, serial_number, written_on, quantity_dispende, date_billed
188 start transaction;
189 do
190 $$
191 begin
192     call add_prescription('diclofenac epolamine','Stark', 'Tony', cast('29-may-1970' as date), 'Banner', 'Bruce', 1, 'eeeeeeee', cast('15-oct-2022' as date), 90, cast('15-oct-2022' as date));
193     call add_prescription('restasis','Stark', 'Tony', cast('29-may-1970' as date), 'Banner', 'Bruce', 1, 'eeeeeeee', cast('15-oct-2022' as date), 90, cast('15-oct-2022' as date));
194     call add_prescription('aspirin', 'Stark', 'Tony', cast('29-may-1970' as date), 'Strange', 'Stephen', 1, 'eeeeeeee', cast('10-oct-2022' as date), 90, cast('10-oct-2022' as date));
195     call add_prescription('aspirin', 'Romanoff', 'Natasha', cast('03-dec-1984' as date), 'Strange', 'Stephen', 1, 'eeeeeeee', cast('07-oct-2022' as date), 30, cast('07-oct-2022' as date));
196     call add_prescription('losartan', 'Romanoff', 'Natasha', cast('03-dec-1984' as date), 'Strange', 'Stephen', 1, 'eeeeeeee', cast('09-oct-2022' as date), 30, cast('09-oct-2022' as date));
197     call add_prescription('ibuprofen', 'Odinson', 'Thor', cast('01-jan-0964' as date), 'Doom', 'Victor Von', 1, '999999999', cast('15-oct-2022' as date), 60, cast('15-oct-2022' as date));
198     call add_prescription('minto Plus Tabs', 'Parker', 'Peter', cast('10-aug-2010' as date), 'Octavious', 'Otto', 1, 'FFRX09', cast('15-oct-2022' as date), 90, cast('15-oct-2022' as date));
199     call add_prescription('naproxen DR', 'Parker', 'Peter', cast('10-aug-2010' as date), 'Octavious', 'Otto', 1, 'eeeeeeee', cast('12-oct-2022' as date), 30, cast('12-oct-2022' as date));
200     call add_prescription('jardiance', 'Rogers', 'Steve', cast('04-jul-1920' as date), 'Pym', 'Hank', 1, '999999999', cast('15-oct-2022' as date), 30, cast('15-oct-2022' as date));
201     call add_prescription('amlodipine', 'Rogers', 'Steve', cast('04-jul-1920' as date), 'Doom', 'Victor Von', 1, '999999999', cast('15-oct-2022' as date), 30, cast('15-oct-2022' as date));
202     call add_prescription('omeprazole', 'Rogers', 'Steve', cast('04-jul-1920' as date), 'Pym', 'Hank', 1, '999999999', cast('15-oct-2022' as date), 30, cast('15-oct-2022' as date));
203     call add_prescription('vascepa', 'Fury', 'Nick', cast('04-jul-1950' as date), 'Pym', 'Hank', 1, 'eeeeeeee', cast('14-oct-2022' as date), 120, cast('14-oct-2022' as date));
204 end$$;
205 commit transaction;
```

Below the editor, the 'Statistics' panel shows the execution results. The 'Name' column lists 'patient 1' and 'Statistics 2'. The 'Value' column shows 'Updated Rows 0'. The 'Query' column displays the executed SQL code, which is a subset of the code in the editor. The status bar at the bottom indicates 'EST en\_US Writable', 'Smart Insert', and '188 : 1 [2291] Sel: 2291 | 29'.