ER Management relational schema with update and delete behaviors

Person (ssn, bdate, fname, lname, address)

Staff (sssn [FK1], staffID, hiredate, salary)

Unique attributes: {staffID}

FK1: sssn → Person (ssn)

- *On update restrict:* The ssn of staff should not change. If there is an attempt to change it, this will restrict it.
- *On delete cascade:* If a staff member is fired, we no longer need to keep their information

Department (deptID, dept_name, manager [FK2])

Unique attributes: {manager}

FK2: manager → Staff (sssn)

- *On update cascade:* If a manager is changed, the ssn should be updated for the new manager.
- *On delete restrict:* A department cannot function without a manager, so we would want to restrict this behavior; manager cannot be null.

Room (room_num, room_type, controlling_dept[FK3])

FK3: controlling_dept → Department (deptID)

- *On update cascade:* Many rooms can be managed by one department. Updating the controlling department number is allowed.
- *On delete restrict*: A room must be controlled by a department. An attempt to have a room without a controlling department will be restricted.

Patient (pssn[FK4], contact, funds, occupy_room [FK5])

FK4: pssn → Person(ssn)

- *On update restrict*: The ssn of a patient should not change. If there is an attempt to change it, this will restrict it.
- *On delete restrict:* We need to keep record of all patients even if not in active treatment. If there is an attempt to delete it, this will restrict it.

FK5: occupy_room → Room (room_num)

- *On update cascade*: A patient can change rooms if they are transferred to another department so we would want to update their room number in our records.
- *On delete set null:* A patient does not need to occupy a room. If they no longer need a room, the value will be null.

Doctor (dssn[FK6], licenseNumber, experience)

Unique attributes: {licenseNumber}

FK6: dssn → Staff(sssn)

- *On update restrict*: The ssn of a doctor should not change. If there is an attempt to change it, this will restrict it.
- *On delete restrict:* Even if a doctor is fired, we still want to know what medications they prescribed and patients they treated so this behavior would be restricted.

Orders (orderNumber, priority, order_date, cost, placedby_doctor[FK7], for_patient[FK8])

FK7: placedby_doctor → Doctor (dssn)

- *On update restrict:* One doctor places an order and after it's placed it cannot be undone, so you should not be able to change what doctor made the order.
- *On delete restrict*: an order must be placed by a doctor, so placedby_doctor cannot be null for a given order.

FK8: for_patient → Patient (pssn)

- *On update restrict:* Each order corresponds to one specific patient, and this patient should not be changed. If there is any attempt to change it, it will be restricted.
- *On delete restrict:* An order must be for a patient - an order cannot exist without a patient so this cannot be null and any attempt to delete a patient from an order will be restricted.

Appointment (apptssn [FK9], booking_date, booking_time, cost)

FK9: apptssn → Patient (pssn)

- *On update restrict:* The patient ssn should not be changed. Any attempt to do so will be restricted.
- *On delete restrict*: An appointment cannot exist without a corresponding patient – any attempt to delete a patients ssn will be restricted.

Nurse (nssn[FK10], shiftType, regExpiration)

FK10: nssn → Staff(sssn)

- *On update restrict:* The ssn of nurses should not change. If there is an attempt to change it, this will restrict it.
- *On delete cascade*: if a nurse is fired, we no longer need to keep their information.

Prescription (orderNumber[FK11], drugType, dosage)

FK11: orderNumber → Orders(orderNumber)

- *On update restrict:* Once an order is placed, it cannot be undone so a prescription's order number should not change.
- *On delete cascade:* If a doctor decides the patient does not need this prescription and maybe should have a different prescription, we need to be able to delete the order. If a prescription is deleted, we no longer need to store its information.

LabWork (lab_order_num[FK12], lab_type)

FK12: lab_order_num → Order(orderNumber)

- *On update restrict:* The order number of lab work should not be changed.
- *On delete cascade:* If a doctor decides the patient does not need this lab work or maybe needs different lab work, we need to be able to delete the order. If a lab order is deleted, we no longer need to store its information.

Symptoms ((syssn, booking_date, booking_time)[FK13], sym_type, num_days)

FK13: syssn, booking_date, booking_time → Appointment(apptssn, booking_date, booking_time)

- *On update restrict:* A person's ssn should not be changed. Any attempt to do so will be restricted. A symptom is associated with a certain appointment date and time where they expressed those symptoms so we would not want to update an appointment date or time.
- *On delete restrict:* Even if a patient was not an active patient, we would still want to keep records of their symptoms in case they come back so we would want to restrict deleting patient. You cannot change when a symptom was recorded in the patient's history - the appt associated with the symptom should not be deleted.

Works_in((staff_ssn)[FK14], department[FK15])

FK14: staff_ssn → Staff (sssn)

- *On update restrict:* We would never be changing a staff member's ssn so this behavior would be restricted.
- *On delete restrict:* Even if a staff member left the hospital, we would still want to remember what department they worked in.

FK15: department → Department(deptID)

- *On update cascade:* If a staff member changes departments, the info should be updated to reflect whatever department they are now working in.
- *On delete restrict:* Total participation requires the staff to work in a department, any attempt to delete a department that staff are currently working in would be restricted.

Assigned (nurse_ssn[FK16], room_num[FK17])

FK16: nurse_ssn → Nurse(nssn)

- *On update restrict:* We would never be changing a nurse's ssn, so this behavior would be restricted.
- *On delete restrict:* We would never be deleting a nurse's ssn, so this behavior would be restricted, and a room cannot exist without a nurse so we would not want to delete nurse if they are the only one assigned a room.

FK17: room_num → Room(room_num)

- *On update cascade:* A nurse can be reassigned to a different room so the room number should be updated to show wherever the nurse is currently working.
- *On delete cascade:* Partial participation says a nurse does not need to be assigned a room so rooms can be deleted.

Scheduled_for (doc_ssn[FK18], (appt_booking_time, appt_booking_date, assn)[FK19])

FK18: doc_ssn → Doctor(dssn)

- *On update restrict:* We would never be changing a doctor's ssn, so this behavior would be restricted.
- *On delete restrict:* We would never be deleting a doctor's ssn, so this behavior would be restricted and an appointment cannot exist without a doctor so we would not want to delete doctor if they are the only one assigned an appointment.

FK19: appt_booking_time, appt_booking_date, assn → Appointment(booking_date,

booking_time, apptssn)

- *On update cascade:* An appointment time and date should be able to change in cases of rescheduling.
- *On delete cascade:* An appointment time and date should be able to be deleted in cases of cancellation.