

## 1. 查询：通过就诊编码和状态查询患者路径 patcp/query/visitCodeAndStatusCode

- 会查出所有患者路径相关的信息，所有的临床路径相关的表
- 是否进入下一步弹窗

## 2. 诊断开立

## • 逻辑：

- 根据患者visit\_code和is\_chief=1(主诊断)条件查诊断表所有数据
  - 将所有非医保的主诊断信息，构建map,key=诊断编码，value=诊断实体
- 查询所有状态为ACTIVE下的临床路径：cp/query/status
- 循环比对：
  - 路径停用时间超出当前时间则过滤
  - CP\_VS\_DESC表中CP\_VS\_SUB\_TYPE（对应子类）为选中的诊断类型，字典（InCPType）
  - CP\_VS\_DESC表中CP\_VS\_TYPE（对应类别）取值为6中的cpDescType，为table也的编号。
  - （进入路径标准 && 中医或西医诊断 && map能取到） || 诊断校验是否同类（西医肺炎）
  - 判断代码：

```

        if (pathways != null && pathways.length > 0) {
        for (ClinicalPathwayVO cp : pathways) {
            Date patRcmc = getRestService().getSysDate();//服务端当前时
间
            List<CPvsDescVO> cpDescList = cp.getCpDescList();
            if (cpDescList == null) {
                continue;
            }
            if (cp.getEffectiveTimeHigh() != null
                && patRcmc.after(cp.getEffectiveTimeHigh())) { //停
用时间超出当前时间则过滤
                continue;
            }
            for (CPvsDescVO cpDesc : cpDescList) {
                if (("1".equals(cpDesc.getCpVsType())
                    && ("1".equals(cpDesc.getCpVsSubType())
                    || "3".equals(cpDesc.getCpVsSubType()))
                    && diagMap.get(cpDesc.getCpVsCode()) != null)
                    || diagnosisCheck(cpDesc, clinicDiagnosis)) {
                    patCP.add(cp);
                    break;
                }
            }
        }
    }

    /**
     * 诊断校验是否同类（肺炎）
     *
     * @param cpDesc
     * @return
     */
    private boolean diagnosisCheck(CPvsDescVO cpDesc,
        ClinicDiagnosis[] clinicDiagnosis) {

```

```

        for (ClinicDiagnosis clinicDiagnosi : clinicDiagnosis) {
            if (cpDesc.getCpVsCode() != null
                && clinicDiagnosi.getDiagnosisCode() != null
                && (cpDesc.getCpVsCode().contains("J13")
                    || cpDesc.getCpVsCode().contains("J14")
                    || cpDesc.getCpVsCode().contains("J15")
                    || cpDesc.getCpVsCode().contains("J18")))
                &&
                (clinicDiagnosi.getDiagnosisCode().contains("J13")
                    ||
                    clinicDiagnosi.getDiagnosisCode().contains("J14")
                    ||
                    clinicDiagnosi.getDiagnosisCode().contains("J15")
                    ||
                    clinicDiagnosi.getDiagnosisCode().contains("J18"))) {
                return true;
            }
        }
        return false;
    }
}

```

### 3. 进入路径: /patcp/enterCP

- 条件:
  - 在院表CP\_STATUS\_CODE为ACTIVE、ABORTED、COMPLETED状态不允许进入路径
  - 患者入径后, 不受升级操作影响, 仍按照原有路径维护数据操作
- 表操作:
  - 将PAT\_IMP\_IN\_HOSPITAL和PAT\_IMP\_INPATIENT\_LIST表中的CP\_STATUS\_CODE更新为ACTIVE
  - 设置页面上方患者信息: 路径总天数、当前步骤、当前第几天 (取CP\_PAT表, 入路径后设置CpDays=1)
  - 插入CP\_PAT\_V2 (操作字段 EXECUTING中 入径时间)
  - 插入CP\_PAT\_ORDER表, 数据取CP\_STEP\_ITEM表中所有此路径步骤项目, 状态NEW  
CP\_ORDER\_ID为路径步骤项目id, 开立后ORDER\_ID为通过路径开立时候医嘱的id (现为NULL)
  - 插入CP\_PAT
  - 数据流

```

update PAT_IMP_IN_HOSPITAL SET CP_STATUS_CODE = 'ACTIVE'
WHERE VISIT_CODE = '1312852';

```

```

update PAT_IMP_INPATIENT_LIST SET CP_STATUS_CODE = 'ACTIVE'
WHERE VISIT_CODE = '1312852';

```

```

insert into CP_PAT_V2 ( ID, VISIT_CODE, TEMP_ID, BEGIN_TIME,
STATUS_CODE ) values ( 1745, '1312851', 2677, '2021-12-22 17:28:07.0',
'EXECUTING' );

```

```

insert into CP_PAT_ORDER ( ID, VISIT_CODE, CP_ID, CP_CODE, CP_STEP_ID,
CP_ORDER_ID, STATUS_CODE ) values ( 609906, '1312851', 2677,
'CP00002845', 3866, 53677, 'NEW' );

```

```

insert into CP_PAT_ORDER ( ID, VISIT_CODE, CP_ID, CP_CODE, CP_STEP_ID,
CP_ORDER_ID, STATUS_CODE ) values ( 609907, '1312851', 2677,
'CP00002845', 3866, 53678, 'NEW' );

insert into CP_PAT_ORDER ( ID, VISIT_CODE, CP_ID, CP_CODE, CP_STEP_ID,
CP_ORDER_ID, STATUS_CODE ) values ( 609908, '1312851', 2677,
'CP00002845', 3867, 53675, 'NEW' );

insert into CP_PAT_ORDER ( ID, VISIT_CODE, CP_ID, CP_CODE, CP_STEP_ID,
CP_ORDER_ID, STATUS_CODE ) values ( 609909, '1312851', 2677,
'CP00002845', 3867, 53676, 'NEW' );

insert into CP_PAT ( ID, VISIT_CODE, CP_ID, CP_CODE, CP_NAME,
CP_VERSION, CP_ORG_CODE, IN_CP_DOCTOR, IN_CP_DATE, EFFECTIVE_TIME_LOW,
EFFECTIVE_TIME_HIGH, STATUS_CODE, CP_DAYS, NOW_CP_STEP_ID,
NEXT_CP_STEP_ID, DIAGNOSIS_CODE, DIAGNOSIS_NAME ) values ( 6431,
'1312851', 2677, 'CP00002845', '测试三', 'V1', '0001102102', '009999',
'2021-12-22 17:28:07.0', '2021-12-22 17:28:07.0', '2021-12-26
17:28:07.0', 'ACTIVE', 1, 3866, 3867, 'J39.300', '上呼吸道过敏反应' );

```

#### 4. 勾选开立医嘱：patcp/insertOrderByStepItems

- 开立条件判断：

- CPStopImpOrdersCommand
- 路径开始时间不能为空
- 不能同时开立多条分级护理医嘱
- 只能开立当前步骤的医嘱项目
- 预出院或预转科患者（在院表状态为ABORTED）禁止开立医嘱
- 出院患者（在院表状态为COMMAND）禁止开立医嘱
- 路径出院、会诊医嘱必须单独开立
- 开关Clinic\_OutDiagnosisRequired开立 " 出院 " 类型医嘱时是否限制必须有出院诊断。查诊断信息
- 同类药判断(逻辑有误待屏蔽)
- 替换药药品和服务项目编码，在开立的时候替换掉
- 抗菌药物的治疗和预防（弹窗填写）
- 针对开立中成药、西药长期项目，判断开关CLINIC\_LONG\_ORDER\_BAN\_DISPENSARY（住院长期医嘱禁止使用的药房，未维护为不限制），判断所勾选项目的执行科室是否在开关中维护，是则拦截
- 校验药品库存（执行科室）
- 校验药品可不可拆
- 服务项目是否在用（状态 适用范围）
- 性别限制开立项目
- 停止长期（手术、术后）医嘱数据处理，作废临时医嘱数据处理
- 出院、术后、手术类弹出停止长期医嘱窗口
- 手术、术后、出院、转科，弹停止医嘱
- insertImpPatOrdersByCPCommand
- 校验处方权、毒药、麻醉、精神一、二药品、贵重药品、抗肿瘤药品权限
- 校验皮试：开关Clinic\_SkinTestResults "皮试结果判断 0 不判断， 1判断"

- 患者已开立医嘱列表中已存在分级护理、出院医嘱不允许重复开立
- 检查病例摘要和诊断（查询已开立医嘱列表中检查医嘱的临床病史，没有就弹窗批量修改，再开立）
- 过滤住院医嘱项目
- 数据流：
  - 新增：patcp/insertOrderByStepItems
  - 更新CP\_PAT\_ORDER中的ORDER\_ID和状态：

```
update CP_PAT_ORDER SET ORDER_ID = 5060952, STATUS_CODE = 'COMPLETED'
WHERE ID = 609907;
```

- 医生站删除径内医嘱：会将CP\_PAT\_ORDER状态改回NEW,可再次勾选开立

```
update CP_PAT_ORDER set VISIT_CODE = '1312852', CP_ID = 2679, CP_CODE =
'CP00002847', CP_STEP_ID = 3869, CP_ORDER_ID = 53690, ORDER_ID = null,
ORDER_SERVICE_CODE = null, DRUG_GOODS_NAME = null, DRUG_GOODS_CODE =
null, STATUS_CODE = 'NEW'
WHERE ID = 609920;
```

## 5. 路径下一步

- 条件判断：
  - 查询路径相关信息：patcp/query/visitCodeAndStatusCode，取CP\_PAT患者路径主表中的入路径时间，当前步骤id
  - 计算时间，判断开关Clinic\_MultiOperation（路径患者一个自然日内允许多次进行路径"下一步"）
  - 查询当前步骤项目信息：cp/step/item/query/stepIdAndType
  - 拿全部与当前步骤比对 必选的医嘱是否勾选，其他非描述类步骤项目（弹窗用）。
- 下一步初始化：commitNextDayCommand
  - 查询所有变异原因：cp/vr/query/all，对应路径变异原因表CP\_VARIATION\_REASON
  - 查询:患者visitCode初始化退出路径页面实体patcp/select/pat/assessment/visitCode
  - 路径是否变异判断逻辑：
    - 将入路径时间、当前时间转换为一年中第多少天，DAYS=当前时间-入路径时间 例：DAYS=358-357
    - 取CP\_PAT表中的路径天数 CP\_DAYS == DAYS？ 正常：变异
  - 查询CP\_PAT和路径实际费用：用当前步骤已开立医嘱order\_id和visit\_Code查住院费用明细表
    - 数据流：econinpatient/queryordercharge

```
select eic.*,dia.approval_user as grantUser,dia.approval_date as
grantDate,dia.send_user as applyUser,dia.send_date as applyDate
FROM (
SELECT *
FROM econ_inpatient_charge charge
WHERE charge.visit_code ='1312852' and charge.order_id =5060969
order by
```

```

charge.execute_date,charge.create_date,charge.price_item_type
desc) eic
LEFT JOIN (select min(d.approval_user) as
approval_user,min(d.approval_date) as
approval_date,min(d.send_user) as send_user,min(d.send_date) as
send_date,d.apply_serial_no
FROM DRUG_IMP_APPLY d group by d.apply_serial_no) dia
ON eic.apply_detail_id = dia.apply_serial_no and
eic.system_item_type='3';

```

- 进入下一步: commitNextDayCommand
  - patcp/next/day
  - 插入评估记录: CP\_PAT\_ASSESSMENT
    - 勾选的变异原因 和 EndCpDate
  - 正常进入下一步
    - 当前路径持续天数小于当前步骤天数时, 只增加天数不做其他操作
    - 当前路径天数等于当前步骤加之前所有步骤天数之和时, 进入下一步骤
    - 更新患者路径主表 CP\_PAT
  - 正常到期完成退径
    - 更新CP\_PAT表的STATUS\_CODE为COMMAND
    - 更新住院表和在院表CP\_STATUS\_CODE为COMMAND
    - 插入正常退径评估信息 CP\_PAT\_ASSESSMENT
    - 执行退径相关表操作:
      - 更新患者主表 CP\_PAT\_v2 状态COMMAND endtime
      - 插入评估表 CP\_ASSESSMENT\_V2
      - 插入变异表 CP\_VARIATION\_V2 原因: 正常退径
      - 插入CP\_PAT\_EXT\_V2 (用于保存结束路径时一些业务上没办法记录的数据, 例如: 手术部位是否感染, 是否重返手术室信息)
  - 正常下一步还是正常退径判断逻辑:
    - 计算当前步骤的总天数 (步骤表中的最大天数MAXNODEDAY) defDuringDays
    - 查CP\_PAT表中的路径天数cpdays
    - 上面两者比较大小
    - 小 证明没有到下一步骤, 把CP\_PAT天数CP\_DAYS加一
    - 等于 证明到了下一步骤, 把CP\_PAT的NOW\_STEP\_ID和NEX\_STEP\_ID更新, CP\_PAT的CP\_DAYS同样加一 (能进入下一步说明天数等于之前所有步骤项目天数相加总和)
    - 点击下一步按钮: CP\_PAT的CP\_DAYS加一, 一天天的加
    - 代码:

```

//正常进入下一步
CPPatV0 gotoNextStep = this.gotoNextStep(cpPatV0);
//正常到期完成

```

```

if
(StatusEnum.COMPLETED.name().equals(gotoNextStep.getStatusCode()))
{
    CPPat condition = new CPPat();
    condition.setId(cpPatVO.getId());
    condition.setStatusCode(StatusEnum.COMPLETED.name());
    cpPatDao.updateByPrimaryKeySelective(condition);
    InHospital patCondition = new InHospital();
    patCondition.setVisitCode(cpPatVO.getVisitCode());
    patCondition.setCpStatusCode(StatusEnum.COMPLETED.name());
    cpOrderFacade.updateInHospital(patCondition);
    InPatientVO inPatient = new InPatientVO();
    inPatient.setVisitCode(cpPatVO.getVisitCode());
    inPatient.setCpStatusCode(StatusEnum.COMPLETED.name());
    cpOrderFacade.updateInPatient(inPatient);
    /*下一步退径数据库插入*/
    CPPatAssessment cpPatAssessment = HipBeanCopier.copy(cpPatVO,
CPPatAssessment.class);
    cpPatAssessment.setEndCpDate(hipDao.selectDataBaseTime());

    cpPatAssessment.setAssessmentDate(hipDao.selectDataBaseTime());
    cpPatAssessment.setAssessmentRemark("下一步完成路径");
    cpPatAssessment.setAssessmentUser(cpPatVO.getInCpDoctor());
    cpPatAssessment.setVariationReason("正常退径");
    cpPatAssessmentDao.insert(cpPatAssessment);
    CAssessmentV2VO cpAssessmentV2VO =
HipBeanCopier.copy(cpPatAssessment, CAssessmentV2VO.class);
    cpAssessmentV2VO.setIsVariation(0);
    cpAssessmentV2VO.setCpPatId(cpPatVO.getCpId().longValue());
    patOutCPV2(cpAssessmentV2VO);
}

//判断方法
private CPPatVO gotoNextStep(CPPatVO cpPatVO) {
    List<CPStepVO> stepList =
cpPatVO.getCpVO().getCpStepList();
    stepList.sort(new Comparator<CPStepVO>() {
        @Override
        public int compare(CPStepVO o1, CPStepVO o2) {
            return (int) (o1.getCreateDate().getTime() -
o2.getCreateDate().getTime());
        }
    });

    //计算到当前步骤总天数
    short defDuringDays = 0;
    for (CPStepVO cpStepVO : stepList) {
        defDuringDays += cpStepVO.getNodeMaxDays();
        if
(cpStepVO.getId().equals(cpPatVO.getNowCpStepId().longValue())) {
            break;
        }
    }
}

```

```

        CPPat condition = new CPPat();
        condition.setId(cpPatVO.getId());
        if (cpPatVO.getCpDays() < defDuringDays) {
            //当前路径持续天数小于当前步骤天数时，只增加天数不做其他操
作
            condition.setCpDays((short) (cpPatVO.getCpDays() +
1));
        } else {
            //当前路径天数等于当前步骤加之前所有步骤天数之和时，进入下
一步骤
            if (cpPatVO.getNextCpStepId() != null
                &&
!cpPatVO.getNextCpStepId().equals(cpPatVO.getNowCpStepId())) {
                condition.setNowCpStepId(cpPatVO.getNextCpStepId());

                Integer nextStepId = null;
                for (int i = 0; i < stepList.size(); i++) {
                    CPStep cpStep = stepList.get(i);
                    if
(cpStep.getId().equals(cpPatVO.getNextCpStepId().longValue())
                        && i < stepList.size() - 1) {
                        nextStepId = stepList.get(i +
1).getId().intValue();
                        break;
                    }
                }
                condition.setNextCpStepId(nextStepId);
                condition.setCpDays((short) (cpPatVO.getCpDays() +
1));
            } else {

                condition.setStatusCode(StatusEnum.COMPLETED.name());
            }
        }

        condition.setEffectiveTimeLow(hipDao.selectDataBaseTime()); //进入下
一天更新
        cpPatDao.updateByPrimaryKeySelective(condition);

        if (condition.getNowCpStepId() != null) {
            cpPatVO.setNowCpStepId(condition.getNowCpStepId());
        }
        if (condition.getNextCpStepId() != null) {
            cpPatVO.setNextCpStepId(condition.getNextCpStepId());
        }
        if (condition.getStatusCode() != null) {
            cpPatVO.setStatusCode(condition.getStatusCode());
        }
        if (condition.getCpDays() != null) {
            cpPatVO.setCpDays(condition.getCpDays());
        }
        return cpPatVO;
    }

```

- 数据流:

- 正常下一步:

- 当前路径持续天数小于当前步骤天数时，只增加天数不做其他操作

```
insert into CP_PAT_ASSESSMENT ( ID, VISIT_CODE, CP_ID,
CP_CODE, END_CP_DATE, EXE_CP_DAYS, ASSESSMENT_USER,
ASSESSMENT_DATE, ASSESSMENT_RESULT, VARIATION_REASON,
VARIATION_TYPE ) values ( 8007, '1312852', 2679,
'CP00002847', '2021-12-24 10:22:15.0', 1, '009999', '2021-12-
24 10:22:15.0', '2', '家属闹事', '2' );
```

```
update CP_PAT SET EFFECTIVE_TIME_LOW = '2021-12-24
10:22:15.0', CP_DAYS = 2
WHERE ID = 6436;
```

- 当前路径天数等于当前步骤加之前所有步骤天数之和时，进入下一步骤

```
insert into CP_PAT_ASSESSMENT ( ID, VISIT_CODE, CP_ID,
CP_CODE, END_CP_DATE, EXE_CP_DAYS, ASSESSMENT_USER,
ASSESSMENT_DATE, ASSESSMENT_RESULT, VARIATION_REASON,
VARIATION_TYPE ) values ( 8008, '1312852', 2679,
'CP00002847', '2021-12-24 10:35:30.0', 2, '009999', '2021-12-
24 10:35:30.0', '2', '病区好转', '10' );
```

```
update CP_PAT SET EFFECTIVE_TIME_LOW = '2021-12-24
10:35:30.0', CP_DAYS = 3, NOW_CP_STEP_ID = 3870
WHERE ID = 6436;
```

- 正常到期完成退径:

```
update PAT_IMP_IN_HOSPITAL SET CP_STATUS_CODE = 'COMPLETED'
WHERE VISIT_CODE = '1312852';
```

```
update PAT_IMP_INPATIENT_LIST SET CP_STATUS_CODE = 'COMPLETED'
WHERE VISIT_CODE = '1312852';
```

```
insert into CP_PAT_ASSESSMENT ( ID, VISIT_CODE, CP_ID, CP_CODE,
END_CP_DATE, EXE_CP_DAYS, ASSESSMENT_USER, ASSESSMENT_DATE,
ASSESSMENT_RESULT, VARIATION_REASON, VARIATION_TYPE ) values (
8010, '1312852', 2679, 'CP00002847', '2021-12-24 11:26:04.0', 4,
'009999', '2021-12-24 11:26:04.0', '2', '病区好转', '10' );
```

```
update CP_PAT SET EFFECTIVE_TIME_LOW = '2021-12-24 11:26:04.0',
STATUS_CODE = 'COMPLETED'
WHERE ID = 6436;
```



```

update CP_PAT SET STATUS_CODE = 'COMPLETED'
WHERE ID = 6436;

insert into CP_PAT_ASSESSMENT ( ID, VISIT_CODE, CP_ID, CP_CODE,
END_CP_DATE, ASSESSMENT_USER, ASSESSMENT_DATE, ASSESSMENT_REMARK,
VARIATION_REASON ) values ( 8011, '1312852', 2679, 'CP00002847',
'2021-12-24 11:26:04.0', '009999', '2021-12-24 11:26:04.0', '下一步
完成路径', '正常退径' );

update CP_PAT_V2 SET STATUS_CODE = 'COMPLETED'
WHERE ID = 2679;

insert into CP_ASSESSMENT_V2 ( CP_PAT_ID, CP_DAYS, DAYS,
IS_VARIANTION ) values ( 2679, 0, 0, 0 );

```

## 6. 退出路径按钮：patcp/outCP

- 住院、在院表状态：
  - 退出时若患者已开立医嘱CP\_STAUS\_CODE改为ABORTED 不允许再次入径
  - 退出时若患者无医嘱 CP\_STAUS\_CODE改为BACK 此时可以点选择路径按钮再次入路径
- 数据：
  - 更新住院住院表
  - CP\_PAT改为状态同住院表
  - 插入评估表 CP\_ASSESSMENT\_V2
  - 插入变异表 CP\_VARIATION\_V2
  - 插入CP\_PAT\_EXT\_V2 （用于保存结束路径时一些业务上没办法记录的数据，例如：手术部位是否感染，是否重返手术室信息）
  - 数据流：
    - 患者无开立医嘱：

```

update PAT_IMP_IN_HOSPITAL SET CP_STATUS_CODE = 'BACK'
WHERE VISIT_CODE = '1312852';

update PAT_IMP_INPATIENT_LIST SET CP_STATUS_CODE = 'BACK'
WHERE VISIT_CODE = '1312852';

insert into CP_PAT_ASSESSMENT ( ID, VISIT_CODE, CP_ID, CP_CODE,
END_CP_DATE, EXE_CP_DAYS, ASSESSMENT_USER, ASSESSMENT_DATE,
ASSESSMENT_RESULT, VARIATION_REASON, VARIATION_TYPE ) values (
8015, '1312852', 2679, 'CP00002847', '2021-12-24 14:53:53.0', 1,
'009999', '2021-12-24 14:53:53.0', '2', '家属闹事', '2' );

update CP_PAT SET EFFECTIVE_TIME_HIGH = '2021-12-24 14:53:53.0',
STATUS_CODE = 'BACK'
WHERE ID = 6440;

update CP_PAT_V2 SET END_TIME = '2021-12-24 14:53:42.0',
STATUS_CODE = 'VARIATIONOUT'

```

```

WHERE ID = 1756;

insert into CP_ASSESSMENT_V2 ( CP_PAT_ID, CP_COST, COST, CP_DAYS,
DAYS, IS_VARIANTION ) values ( 1756, 0.0, 0.0, 1, 0, 1 );

insert into CP_VARIATION_V2 ( KEY, CP_PAT_ID, TYPE, CATEGORY,
REASON, CREATE_DATE ) values ( '20211224145354', 1756, '-1', '2',
'003', '2021-12-24 14:53:54.0' );

insert into CP_PAT_EXT_V2 (CP_PAT_ID, ITEM_KEY, ITEM_VALUE )
values (1756, 'isOperationInfected', '1' );

```

- 患者有开立医嘱（径内、径外均可）：

```

update PAT_IMP_IN_HOSPITAL SET CP_STATUS_CODE = 'ABORTED'
WHERE VISIT_CODE = '1312852';

update PAT_IMP_INPATIENT_LIST SET CP_STATUS_CODE = 'ABORTED'
WHERE VISIT_CODE = '1312852';

insert into CP_PAT_ASSESSMENT ( ID, VISIT_CODE, CP_ID, CP_CODE,
END_CP_DATE, EXE_CP_DAYS, ASSESSMENT_USER, ASSESSMENT_DATE,
ASSESSMENT_RESULT, VARIATION_REASON, VARIATION_TYPE ) values (
8026, '1312852', 2683, 'CP00002847', '2021-12-24 19:43:39.0', 1,
'009999', '2021-12-24 19:43:39.0', '2', '病情恶化', '10' );

update CP_PAT SET EFFECTIVE_TIME_HIGH = '2021-12-24 19:43:39.0',
STATUS_CODE = 'ABORTED'
WHERE ID = 6446;

update CP_PAT_V2 SET END_TIME = '2021-12-24 19:43:08.0',
STATUS_CODE = 'COMPLETED'
WHERE ID = 1762;

insert into CP_ASSESSMENT_V2 ( CP_PAT_ID, CP_COST, COST, CP_DAYS,
DAYS, IS_VARIANTION ) values ( 1762, 0.0, 0.0, 1, 0, 1 );

insert into CP_VARIATION_V2 ( KEY, CP_PAT_ID, TYPE, CATEGORY,
REASON, CREATE_DATE ) values ( '20211224194339', 1762, '-1', '10',
'001', '2021-12-24 19:43:39.0' );

insert into CP_PAT_EXT_V2 (CP_PAT_ID, ITEM_KEY, ITEM_VALUE )
values (1762, 'isOperationInfected', '1' );

```

## 7. 路径打印：

- 条件：
  - 患者进入过路径才能打印，即在院表CP\_STATUS\_CODE, ACTIVE \ABORTED \ BACK\ COMPLETED 可以打印

- 查医嘱修改数据
- 查路径评估记录
- 勾选的 停止和完成状态的路径项目医嘱

#### 8. 替换同名药、同类药 patcp/replaceDrug

- 同名药：药品名称相同的药
- 同类药：药品药理归类相同的所有有库存的药品
- 给CP\_PAT\_ORDER中的ORDER\_SERVICE\_CODE(替换商品编码)和DRUG\_GOODS\_CODE（替换药品编码）set值，在勾选开立的时候会判断是否有值，有就会用替换药品的信息
- 数据流：

```
update CP_PAT_ORDER set VISIT_CODE = '1312852', CP_ID = 2679, CP_CODE =  
'CP00002847', CP_STEP_ID = 3869, CP_ORDER_ID = 53680, ORDER_ID = null,  
ORDER_SERVICE_CODE = '00000260', DRUG_GOODS_NAME = null,  
DRUG_GOODS_CODE = '00000260', STATUS_CODE = 'NEW'  
WHERE ID = 610016;
```

#### 9. 路径评估：

- 查CP\_PAT\_ASSESSMENT表