**每个表的表格及字段说明**

（1）Taptap游戏评论对应用户信息表（gameToUser）



gameToUser表

关键字段说明

评论id：对游戏评论的唯一标识。

TapTap用户Id:对爬取数据中用户id的存储。

对应游戏id：对爬取数据中游戏id的存储。

用户名称：对爬取数据中游戏的名称的存储。

发布时间：对爬取数据中用户发表评论的时间的存储。

评分星级：对爬取数据中用户对某游戏的评分星级的存储。

游戏时长：对爬取数据中用户玩该游戏的时长的存储。

游戏设备：对爬取数据中用户评论的设备名称的存储

评论内容：对爬取的评论内容的存储

欢乐数：一个轻量级的赞，该评论没有达到赞的程度，却有正面的响应。

点踩数：用户对游戏评论的反面响应。

Taptap游戏信息表（game）



game表

关键字段说明

对应游戏id：对爬取数据中游戏id的存储。

用户名称：对爬取数据中游戏的名称的存储。

简介：对爬取数据中游戏的简介的存储。

最近更新内容：对爬取数据中最近更新内容的存储。

评论数量：对爬取数据中用户评论评论数量的存储。

社区数量：对爬取数据中对该游戏评论的社区数量的存储。

厂商：对爬取数据中游戏厂商的存储。

总评分：对爬取数据中对该游戏的总评分的存储。

类型标签：对爬取数据中游戏的标签的存储。

编辑是否推荐：对爬取数据中对该游戏是否推荐的存储。

下载次数：对爬取数据中用户下载该游戏的次数的存储。

关注数：对爬取数据中用户对该游戏的关注数量的存储。

图片路径：对爬取数据中该游戏的图片存储路径的存储

Taptap用户基本信息表（tapUser）



tapUser表

关键字段说明

用户Id：对爬取数据中用户id的存储。

用户名：对爬取数据中用户名称的存储。

粉丝数：对爬取数据中用户的粉丝数量的存储。

关注数：对爬取数据中用户的关注数的存储。

收藏数：对爬取数据中用户对游戏的收藏数量的存储。

玩过游戏数：对爬取数据中用户玩过的游戏的数量的存储。

玩得最久数：对爬取数据中用户玩过游戏达到一定时长的数量的存储。

评价数：对爬取数据中用户对其他游戏或评论评论的数量的存储。

评分权重：经过评估后，系统给该用户的评价作定性的分数权重。

游戏与用户的多对多映射表（RecentPlay）



映射表

关键字段说明

关联表Id：关联user和game后产生的表，关联Id自增。

游戏Id：对关联的game的id存储。

用户Id：对关联的userd的用户id的存储。

游戏时间：对用户玩该游戏的存储。

应用基本信息表（gameBasic）



gameBasic表

关键字段说明

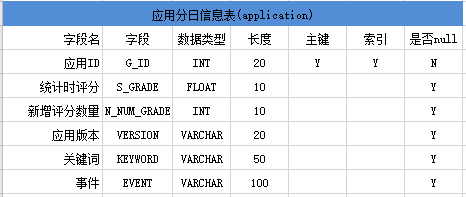
应用Id：对爬取数据中用户id的存储。

应用名称：对爬取数据中用户名称的存储。

最新版评分：应用发布更新后的评分。

权重调整后评分：根据调整后的权重而确定的评分。

应用分日信息表（application）

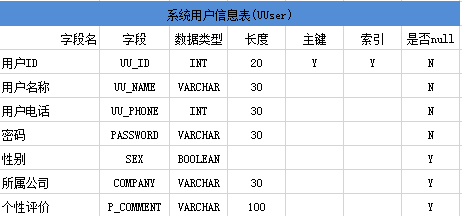


application表

关键字段说明

应用Id：对爬取数据中用户id的存储。

系统用户信息表（UUser）



系统用户表

关键字段说明

用户ID：本系统的用户Id。

用户名称：本系统用户名称。

游戏每日评分缓存（GameDayInfoCache）



评分缓存表

关键字段说明

游戏Id：对爬取数据中游戏id的存储。

缓存信息：对评论数据的时间，星级的整理后的数据。

缓存时间：整理的时间。

关键词缓存表（KeyWordCache）



关键词缓存表

关键字段说明

关键词Id：对分词后的数据存储的ID。

缓存信息：对评论进行分词的数据进行缓存。

缓存时间：整理的时间。

### 数据库方法调用说明

**每个表对应的方法代码说明及活动图**

Taptap游戏评论对应用户信息表（gameToUser）

GameTouserDaoImpl类

类说明：实现GameTouserDao接口，对数据库操作进行事务管理和错误处理

作用：实现对Taptap游戏评论对应用户信息表(gameToUser)的增删改查

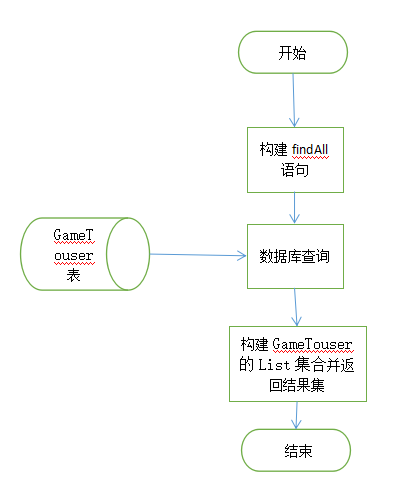
方法说明：

1.public List<GameTouser> findAll()

功能：查询所有gameTouser

参数：无

返回值：GameTouser类的list集合 失败则为null



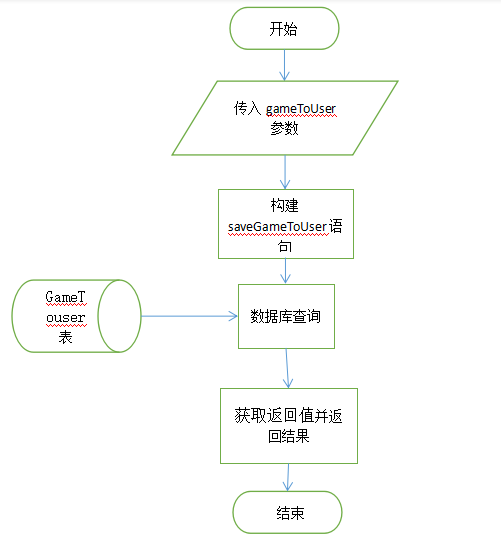
findAll流程图

2.public Integer saveGameTouser(GameTouser gameTouser)

功能：插入一条gameTouser

参数：GameTouser类

返回值：Integer 成功为1，失败为-1



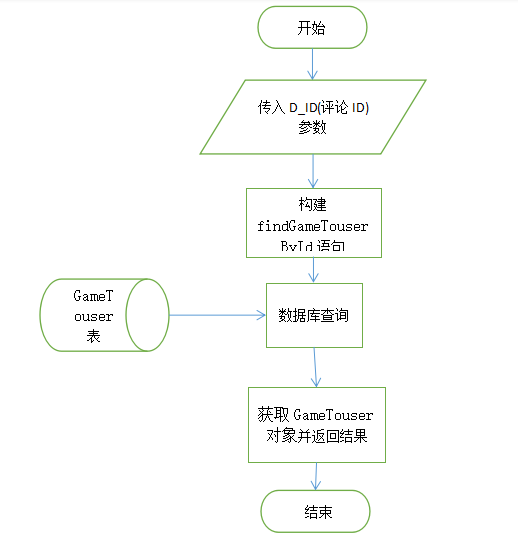
saveGameToUser流程图

3.public GameTouser findGameTouserById(Integer D\_ID)

功能：根据D\_ID（评论id）查询一条gameTouser

参数：D\_ID（评论id）

返回值：GameTosuer类 失败则为null



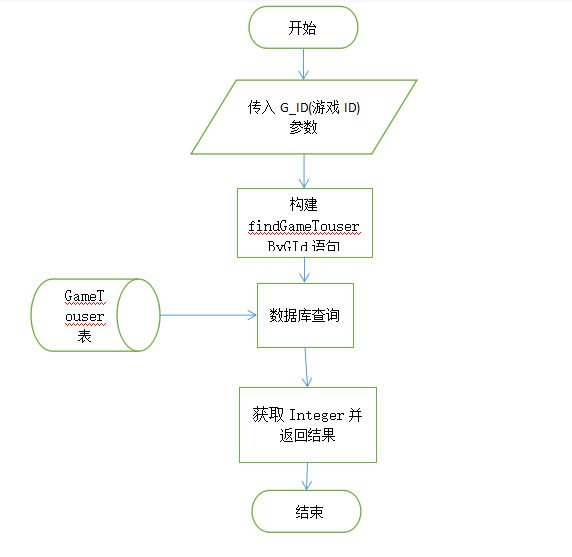
findGameTouserById流程图

4.public Integer findGameTouserByGId(Integer G\_ID)

功能：根据G\_ID（游戏id）查询该游戏评论条数

参数：G\_ID（游戏id）

返回值：Integer 失败则为-1



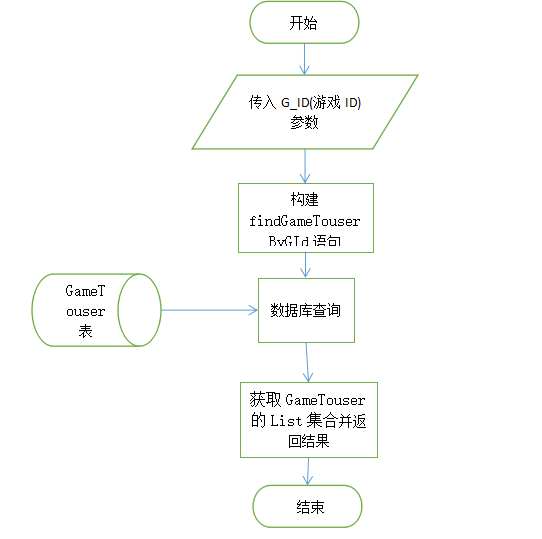
findGameToUserByGId流程图

5.public List<GameTouser> findGameTouserByGId(Integer G\_ID)

功能：根据G\_ID（游戏id）查询一条gameTouser

参数：G\_ID（游戏id）

返回值：GameTosuer的List集合 失败则为null



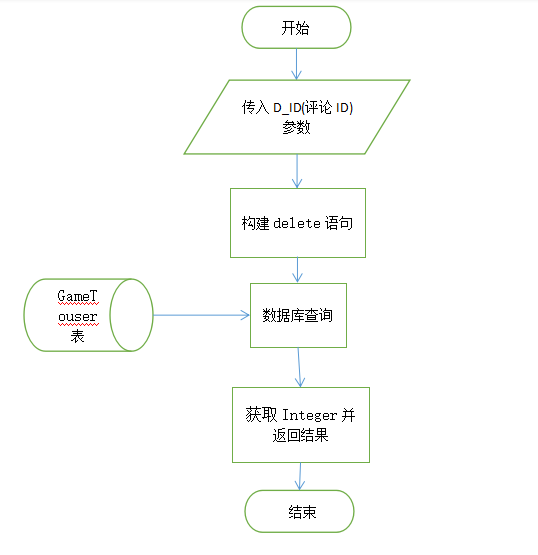
findGameToUserByGId

6.public Integer deleteGameTouser(Integer D\_ID)

功能：根据D\_ID（评论id）删除一条gameTouser

参数：D\_ID（评论id）

返回值：Integer 成功为1，失败为-1



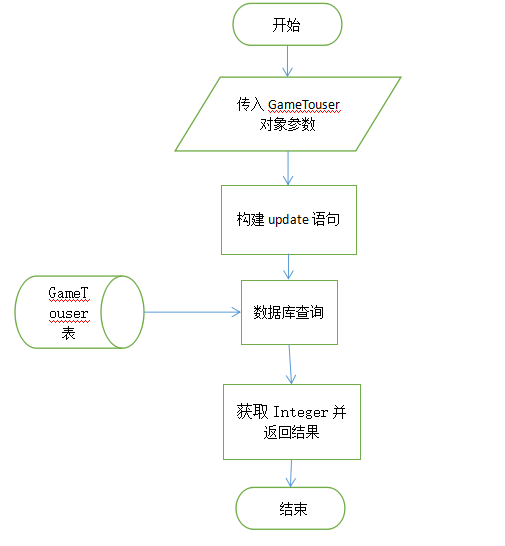
deleteGameToUser流程图

7.public Integer updateGameTouser(GameTouser gameTouser)

功能：更新一条gameTouser

参数：GameTouser类

返回值：Integer 成功为1，失败为-1



updateGameTosuer流程图

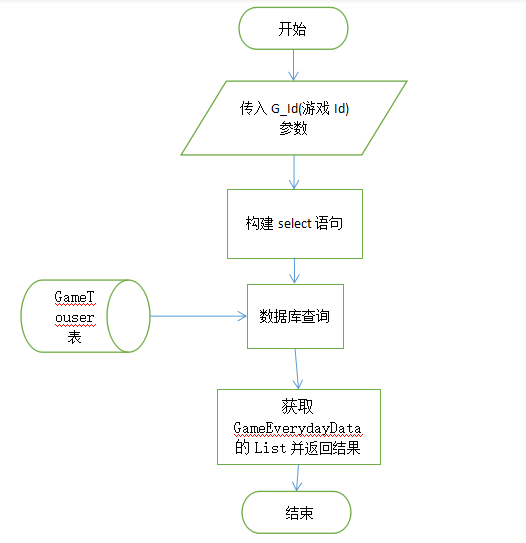
8.public List<GameEverydayData> findCountEverydayAll(Integer G\_ID)

功能：查询每天的评论数量和平均星级

参数：G\_ID（游戏id）

返回值：GameEverydayData类集合 失败则为null

（附GameEverydayData说明：三个属性：time(日期),num（当天评论总数）,average（平均评论星级） 位置：pojo包）



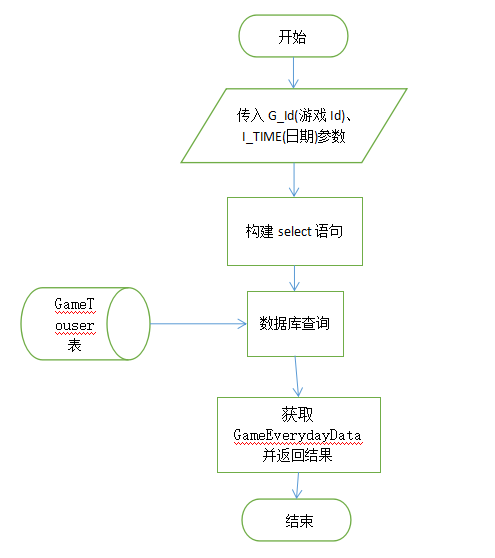
findCountEverydayAll流程图

9.public GameEverydayData findCountByDay(Date I\_TIME,Integer G\_ID)

功能：查询具体某一天的评论数量和评价星级

参数：I\_TIME时间 G\_ID游戏Id

返回值：GameEverydayData 失败为null



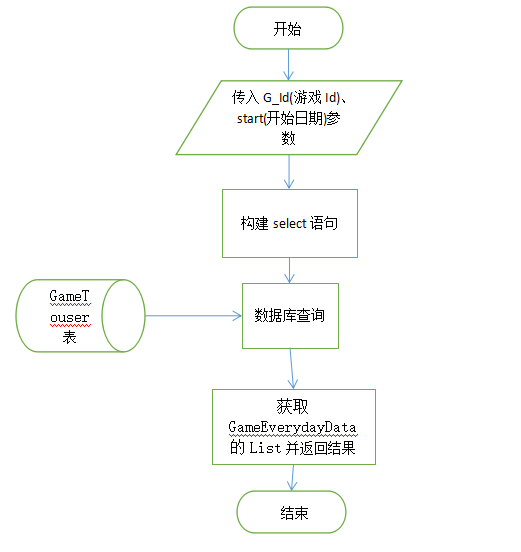
findCountByDay流程图

10.public List<GameEverydayData> findCountAfterDayA(Date start,Integer G\_ID)

功能：查询从某天开始的评论数量和星级

参数：start开始时间 G\_ID游戏Id

返回值：GameEverydayData 失败为null



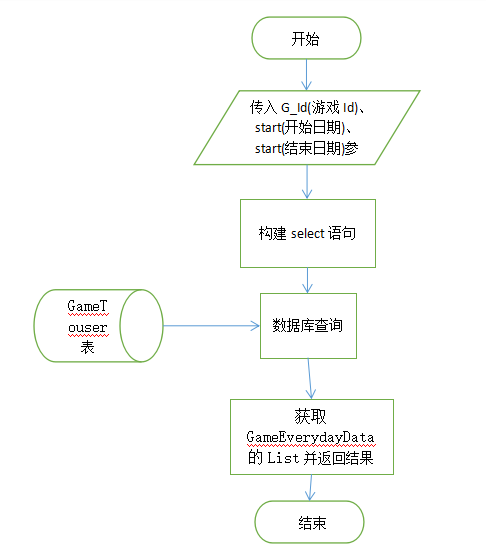
findCountAfterDayA流程图

11.public List<GameEverydayData> findCountBetweenAAndB(Date start,Date en'd，Integer G\_ID)

功能：查询从某天到某天的评论数量和星级

参数：start开始时间 end结束时间 G\_ID游戏Id

返回值：GameEverydayData 失败为null



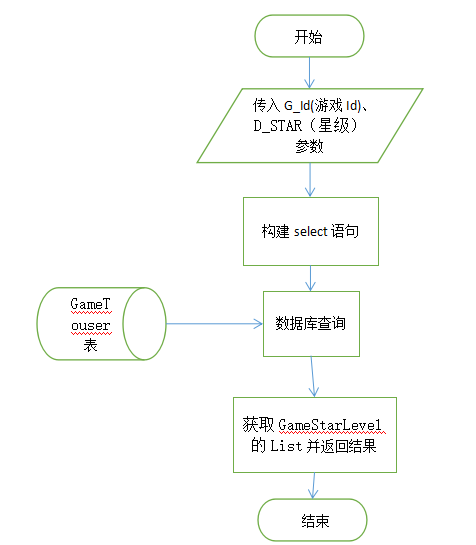
findCountBetweenAAndB流程图

12.public List<GameStarLevel> findCountByStar(Integer D\_START,Integer G\_ID)

功能：查询每天的各星级数量

参数：D\_START星级 G\_ID游戏Id

返回值：GameStarLevel的List集合 失败为null



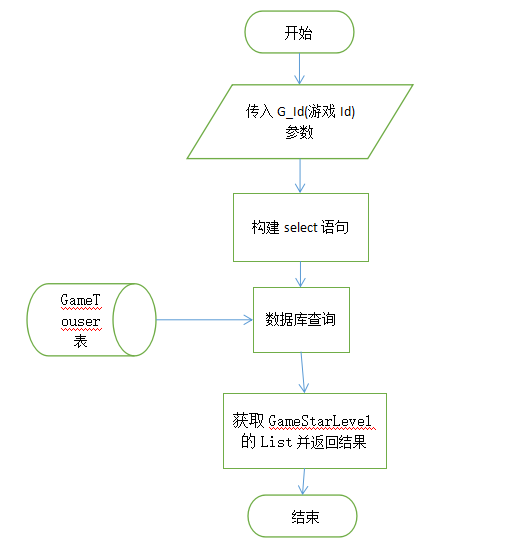
findCountByStar流程图

13.public List<GameStarLevel> findCountByStar(Integer G\_ID)

功能：查询1-5星的每天数量

参数：G\_ID游戏Id

返回值：GameStarLevel的List集合 失败为null



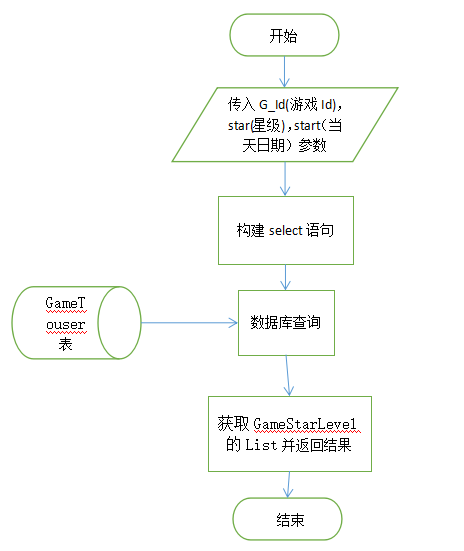
findCountByStar流程图

14.public List<GameStarLevel> findCountStarByDay(Integer G\_ID,Date star,Integer G\_ID)

功能：查询具体一天的各星级数量

参数：star（星级） start(当天日期) G\_ID（游戏Id）

返回值：GameStarLevel的集合 失败为null



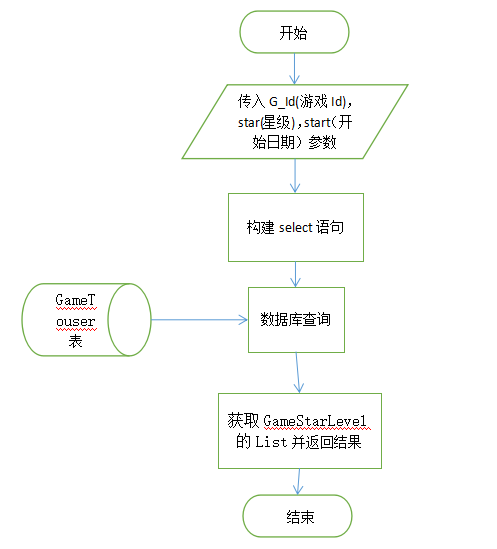
findCountStarByDay流程图

15.public List<GameStarLevel> findCountStarAfterDay(Integer star,Date start,Integer G\_ID)

功能：查询从某天开始的各星级数量

参数：star（星级） start(当天日期) G\_ID（游戏Id）

返回值：GameStarLevel的List集合 失败为null



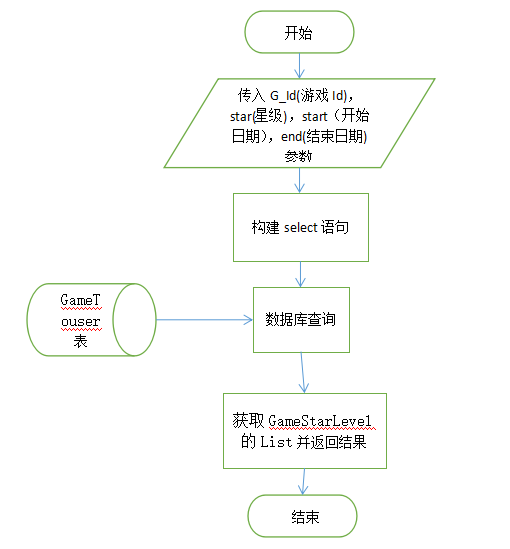
findCountStarAfterDay流程图

16.public List<GameStarLevel> findCountStarBetweenDay(Integer star,Date start,Date end,Integer G\_ID)

功能：查询从某天到某天的各星级数量

参数：star（星级） start(开始日期) end(结束日期) G\_ID（游戏Id）

返回值：GameStarLevel的List集合 失败为null



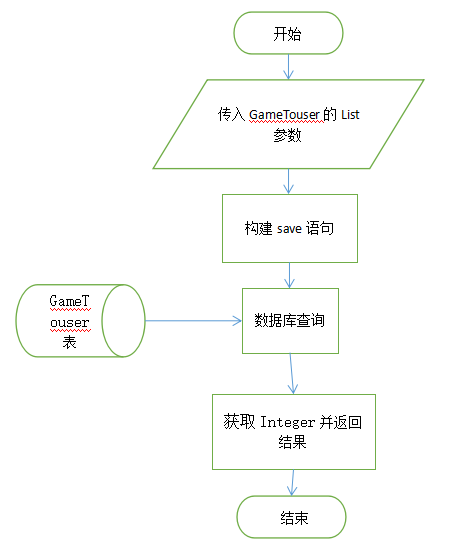
findCountStarBetweenDay

17.public Integer addGameList(List<GameTouser> list)

功能：批量插入批量数据

参数：GameTouser的List集合

返回值：Integer 失败则为-1



addGameList流程图

18.public List<TapUserAndGameToUser> findUnionByGId(Integer G\_ID,Integer U\_ID,String U\_NAME,Integer FANS,Integer ATTENTION,Integer COLLECT,Integer PLAY,Integer L\_PLAY,Float G\_WEIGHT,Integer APPRAISE)

功能：tapUser表与gameToUser表的联合查询,Integer属性不查填-1，float填-1.0f，String为null 字符串为模糊查询

参数：GameTouser的gid和TapUser的所有属性

返回值：TapUserAndGameToUser 失败则为null

1. public List<CompareUser> compareTapAndGameUser();

功能：对TapUser和GamToUser的用户对比，找出TapUser表不存在而GamToUser存在的用户并统计其评论数量

参数：无

返回值：CompareUser(UID和该Id的评论数量)的List 失败为null

20.public List<GameToUser> likeComment(Integer gameId,List<GameToUser> list);

功能：根据List的关键字，对评论的模糊查询,选出排列组合的最多排列的前十的评论

参数：gameId:游戏id,List<String> list:参数的list

返回值：参数的list里选出三个排列组合的最多排列的前十的评论

Taptap游戏信息表（game）

GameDaoImpl类

类说明：实现GameDao接口，对数据库操作进行事务管理和错误处理

作用：实现对Taptap游戏信息表(game)的增删改查

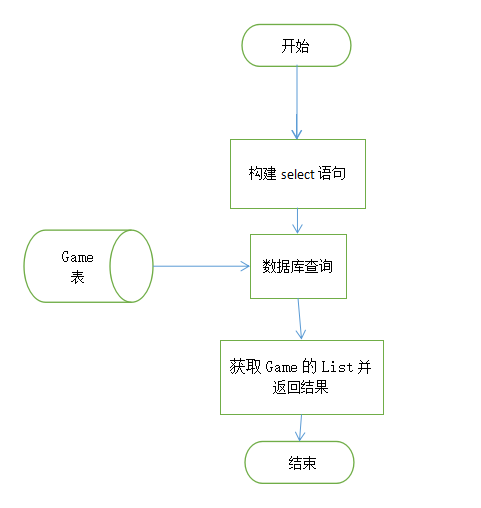
方法说明：

1.public List<Game> findAll()

功能：查询所有game

参数：无

返回值：Game类的list集合 失败则为null



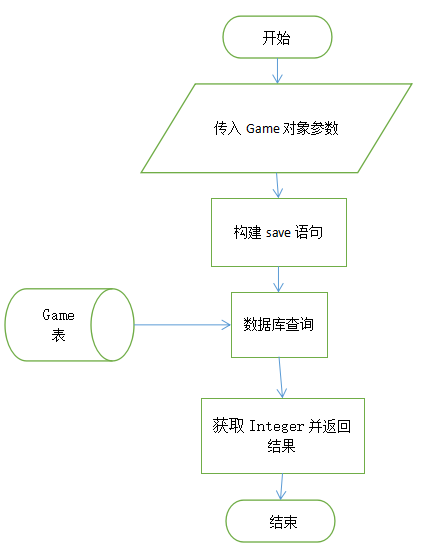
findAll流程图

2.public Integer saveGame(Game game)

功能：插入一条game

参数：Game类

返回值：Integer 成功为1，失败为-1



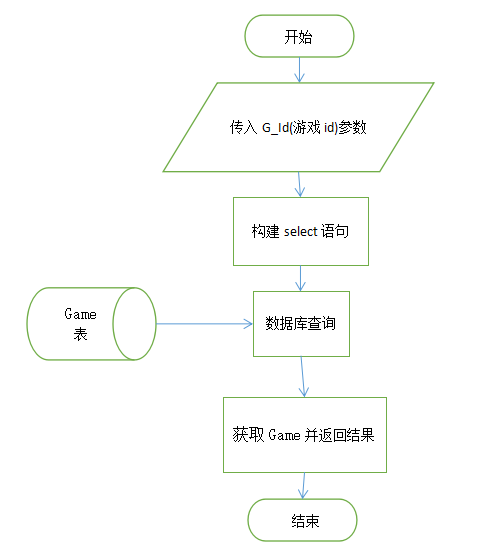
saveGame流程图

3.public Game findGameById(Integer G\_ID)

功能：根据G\_ID查询一条Game

参数：G\_ID（游戏ID）

返回值：Game类 失败则为null



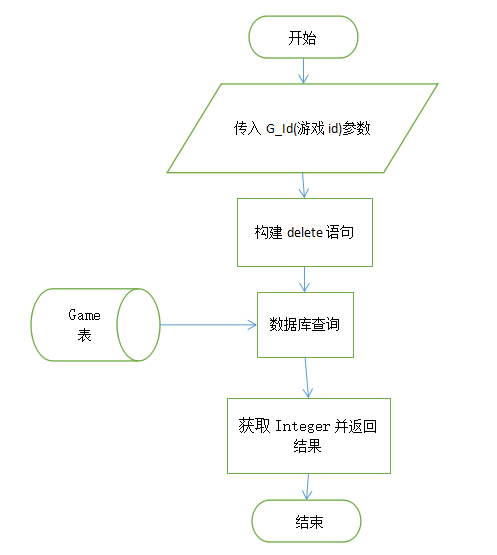
findGameById流程图

4.public Integer deleteGameById(Integer G\_ID)

功能：根据G\_ID删除一条Game

参数：G\_ID（游戏ID）

返回值：Integer 成功为1，失败为-1



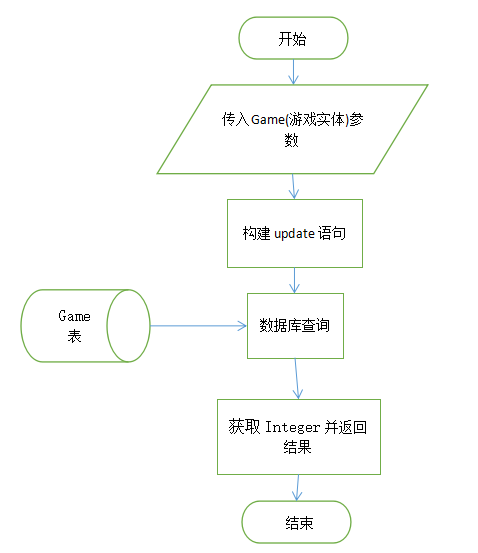
deleteGame流程图

5.public Integer updateGame(Game game)

功能：更新一条Game

参数：Game类

返回值：Integer 成功为1，失败为-1



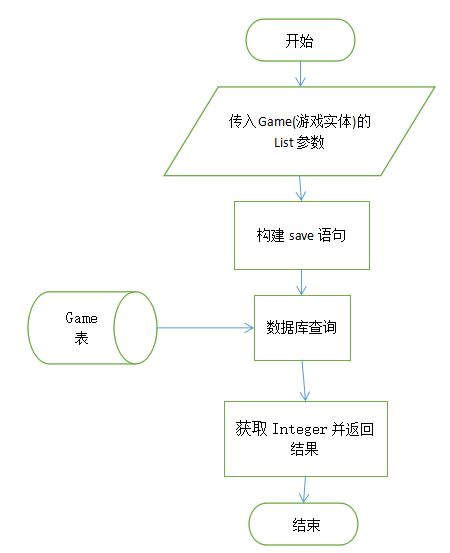
updateGame流程图

6.public Integer saveGame(List<Game> list)

功能：批量插入game

参数：Game类的List

返回值：Integer 失败为-1



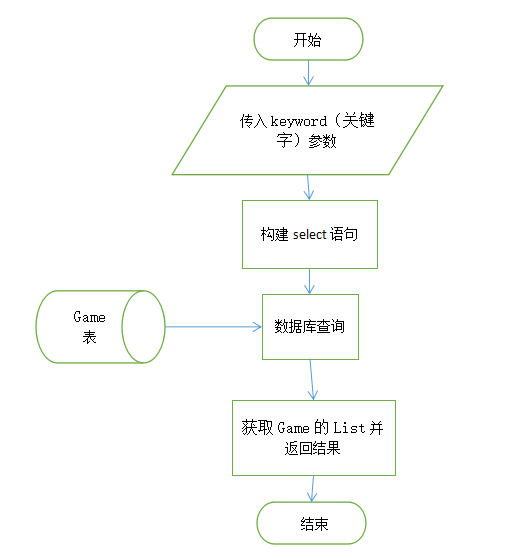
saveGame流程图

7.public List<Game> searchByKeyword(String keyword)

功能：根据关键字检索

参数：keyWord 搜索关键词

返回值：game的List集合 失败为null



searchByKeyword流程图

Taptap用户基本信息表（tapUser）

TapUserDaoImpl类

类说明：实现TapUserDao接口，对数据库操作进行事务管理和错误处理

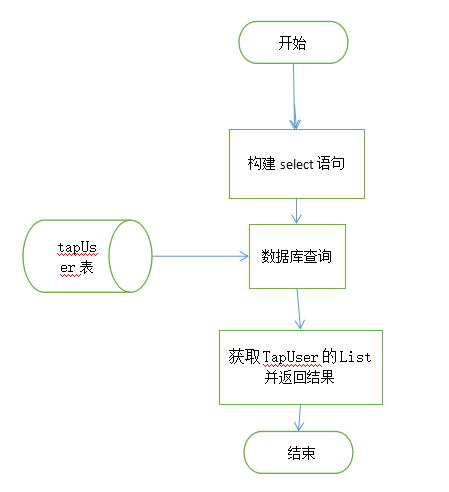
作用：实现对Taptap用户基本信息表（tapUser）的增删改查

方法说明：

1.public List<TapUser> findAll();  
功能：查询所有TapUser

参数：无

返回值：TapUser类的list集合 失败则为null

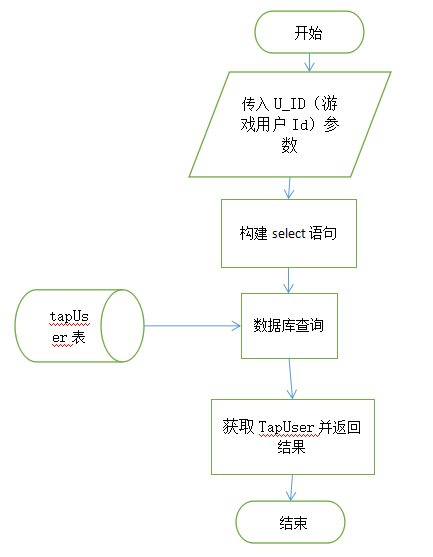


findAll流程图

2.TapUser findTapUserById(Integer U\_ID);  
功能：根据U\_ID查询TapUser

参数：U\_ID（Taptap用户id）

返回值：TapUser类 失败则为null



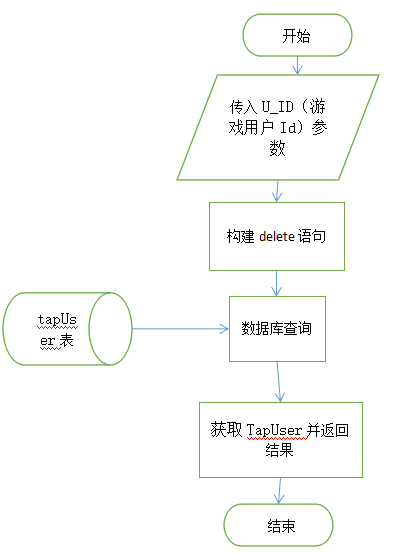
findTapUserById流程图

3.Integer deleteTapUser(Integer U\_ID);

功能：根据U\_ID删除TapUser

参数：U\_ID（Taptap用户id）

返回值：Integer成功为1，失败为-1



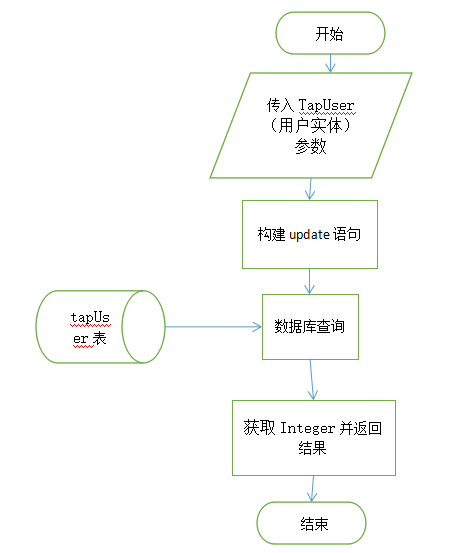
deleteTapUser流程图

4.Integer updateTapUser(TapUser tapUser);

功能：更新TapUser

参数：U\_ID（Taptap用户id）

返回值：Integer成功为1，失败为-1



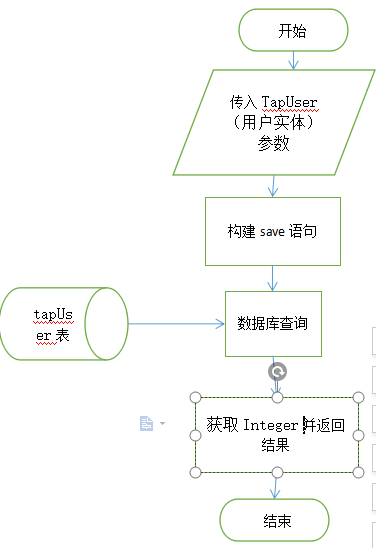
updateTapUser流程图

5.Integer SaveTapUser(TapUser tapUser);

功能：插入一条TapUser

参数：TapUser类

返回值：Integer成功为1，失败为-1



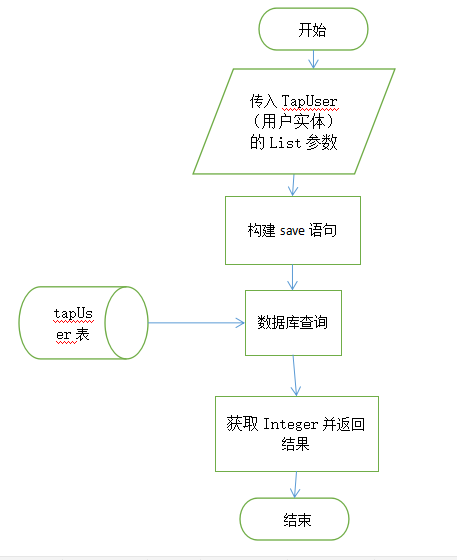
saveTaoUser流程图

6.Integer SaveTapUserList(List<TapUser> list);

功能：批量插入TapUser

参数：TapUser类的集合

返回值：Integer 失败为-1



SaveTapUserList流程图

6.public List<TapUser> findTapUserBeforeDate(Date updateTime)

功能：查询参数时间之前的更新TapUser

参数：Date

参数格式date=new SimpleDateFormat("yyyy-MM-dd").parse("2020-09-26")

返回值：TapUser的List集合 失败为null

游戏与用户的多对多映射表（RecentPlay）

RecentPlayDaoImpl类

类说明：实现RecentPlayDao接口，对数据库操作进行事务管理和错误处理

作用：实现对Taptap游戏与用户的多对多映射表（RecentPlay）的增删改查

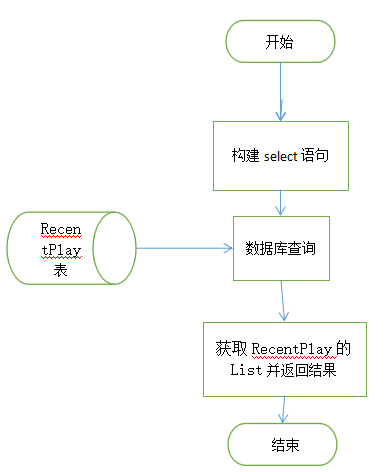
方法说明：

1.public List<RecentPlay> findAll();

功能：查询所有RecentPlay

参数：无

返回值：RecentPlay类的list集合 失败则为null



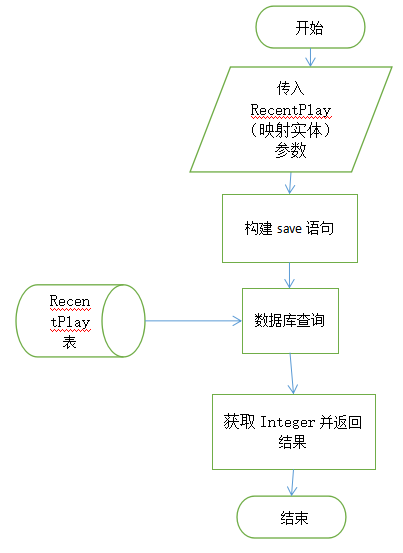
findAllTapUser流程图

2.public Integer saveRecentPlay(RecentPlay recentPlay);

功能：插入一条RecentPlay

参数：RecentPlay类

返回值：Integer成功为1，失败为-1



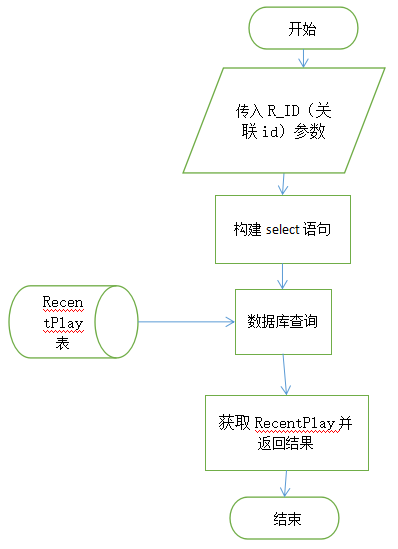
saveRecentPlay流程图

3.public RecentPlay findRecentPlayById(Integer R\_ID);

功能：根据G\_ID查询RecentPlay

参数：R\_ID(关联R\_ID)

返回值：RecentPlay类 失败则为null



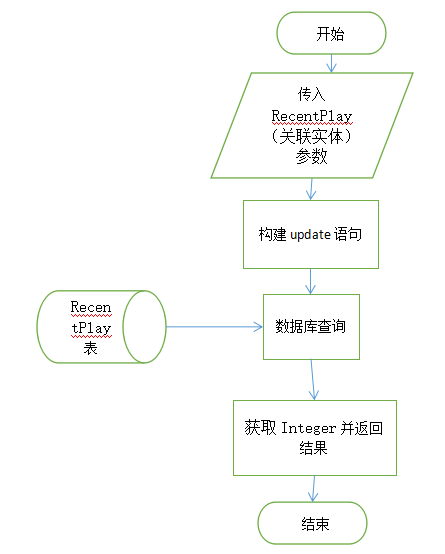
findRecentPlayById流程图

4.public Integer updateRecentPlay(RecentPlay recentPlay);

功能：更新RecentPlay

参数：RecentPlay类

返回值：Integer成功为1，失败为-1



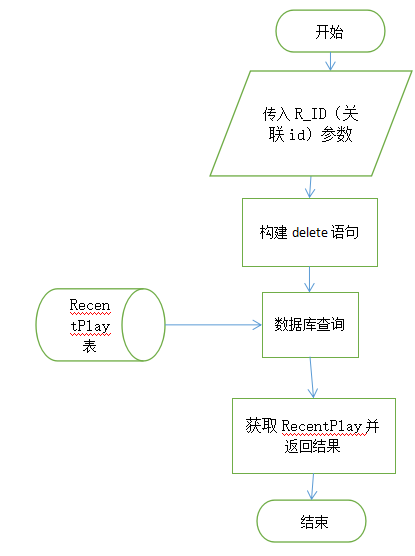
updateRecentPlay流程图

5.public Integer deleteRecentPlayById(Integer R\_ID);

功能：删除RecentPlay

参数：R\_ID(关联ID）

返回值：Integer成功为1，失败为-1



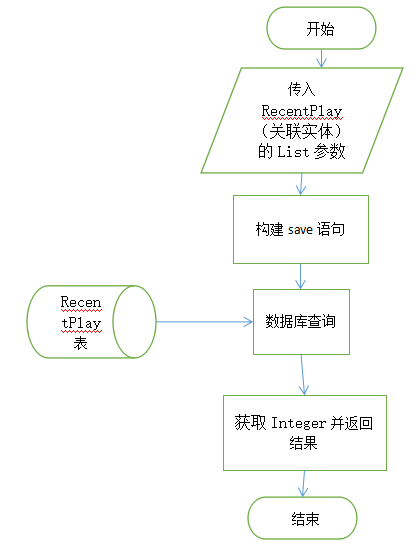
deleteRecentPlayById流程图

6.public Integer saveRecentPlayList(List<RecentPlay> list);

功能：批量插入RecentPlay

参数：RecentPlay类的集合

返回值：Integer成功为插入条数，失败为-1



saveRecentPlayList流程图

应用基本信息表（gameBasic）

GameBasicDaoImpl类

类说明：实现GameBasicDao接口，对数据库操作进行事务管理和错误处理

作用：实现对应用基本信息表（gameBasic）的增删改查

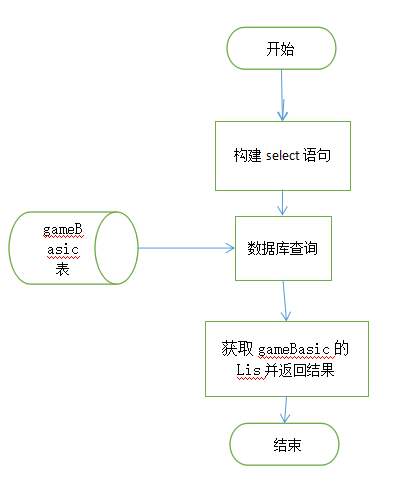
方法说明：

1.public List<GameBasic> findAll();

功能：查询所有GameBasic

参数：无

返回值：GameBasic类的list集合 失败则为null



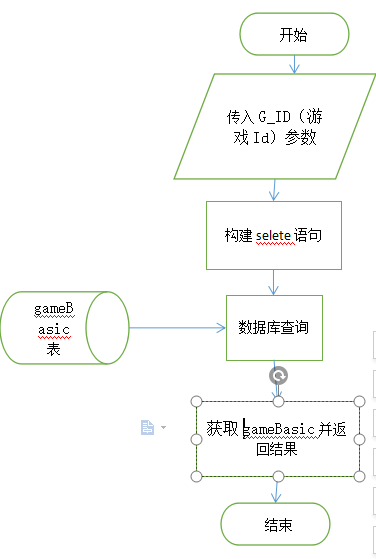
findAll流程图

2.public GameBasic findGBById(Integer G\_ID);

功能：查询一条GameBasic

参数：G\_ID游戏ID

返回值：GameBasic类 失败则为null



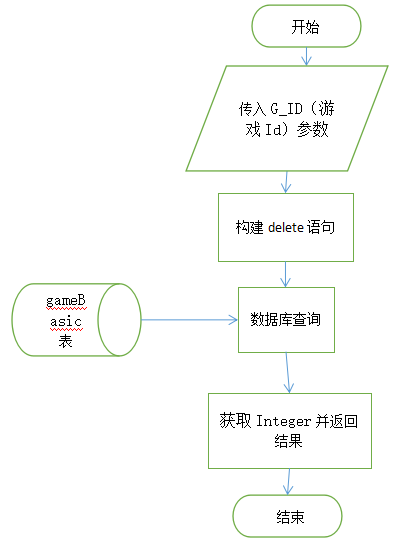
findGBBYId流程图

3.public Integer deleteGameBasic(Integer G\_ID);

功能：删除一条GameBasic

参数：G\_ID游戏ID

返回值：Integer成功为1，失败为-1



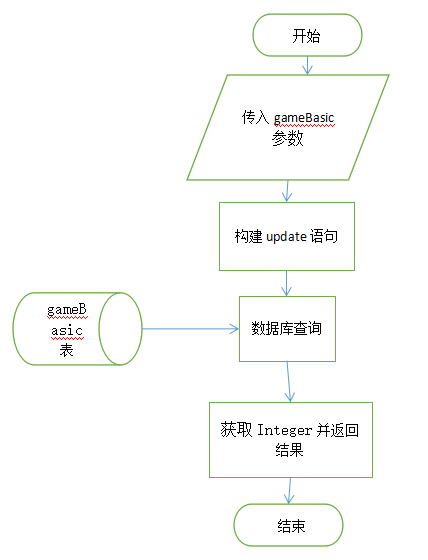
deleteGameBasic流程图

4.public Integer updateGameBasic(GameBasic gameBasic);

功能：更新一条GameBasic

参数：GameBasic类

返回值： Integer 成功为1，失败为-1



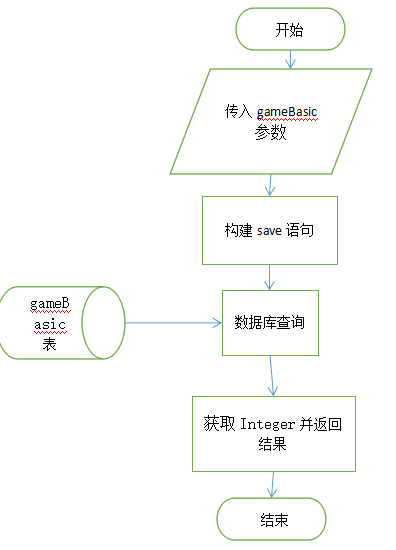
updateGameBasic流程图

5.public Integer saveGameBasic(GameBasic gameBasic);

功能：插入一条gameBasic

参数：GameBasicr类

返回值： Integer成功为1，失败为-1



saveGameBasic流程图

应用分日信息表（application）

ApplicationDaoImpl类

类说明：实现ApplicationDao接口，对数据库操作进行事务管理和错误处理

作用：实现对application（应用分日信息表）表的增删改查

方法说明：

1.public List<Application> findAll()

功能：查询所有application

参数：无

返回值：Application类的list集合

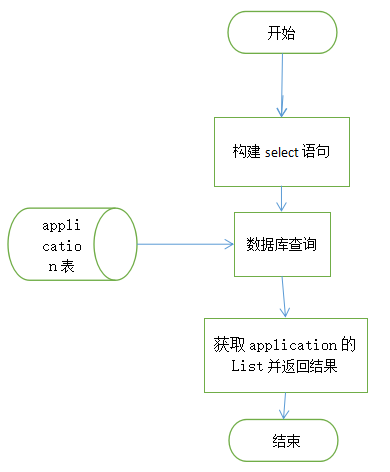


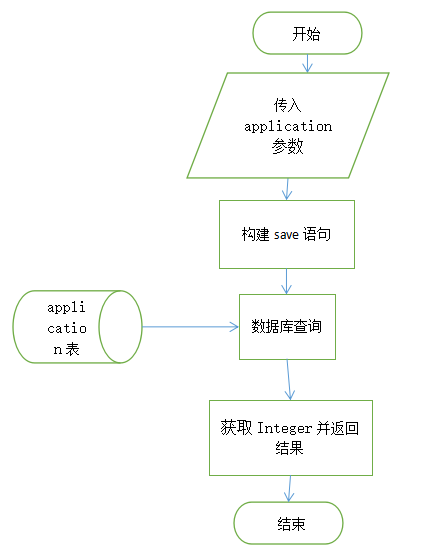
表 1findAll流程图

2. public Integer saveApp(Application application)

功能：插入一条application

参数：Application类

返回值：Integer 成功为1，失败为-1



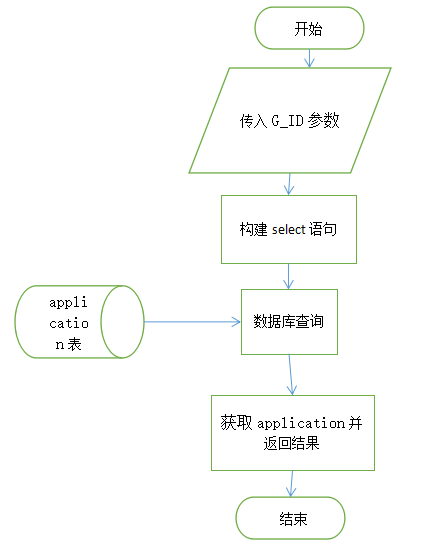
saveApp流程图

public Application findAppById(Integer G\_ID)

功能：根据G\_ID查询一条application

参数：G\_ID（游戏ID）

返回值：Application类 失败则为null



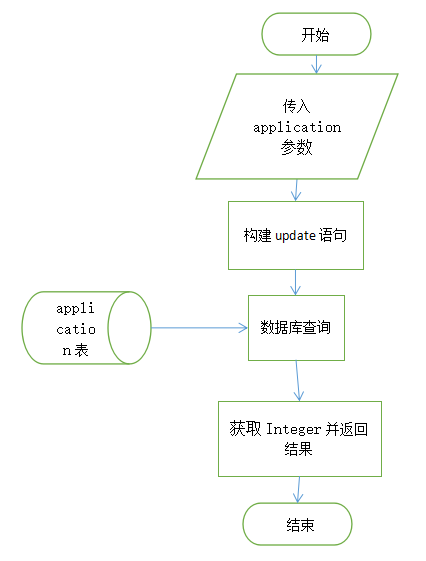
findApplicationById流程图

4.public Integer updateAppByApp(Application application)

功能：对一条application更新

参数：Application类

返回值：Integer 成功为1，失败为-1



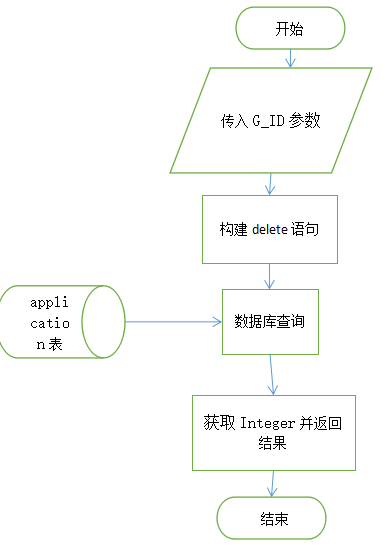
updateApplication流程图

5.public Integer deleteAppById(Integer G\_ID)

功能：根据G\_ID删除一条application

参数：G\_ID(游戏id)

返回值：Integer 成功为1，失败为-1



deleteApplication流程图

系统用户信息表（UUser）

UuserDaoImpl类

类说明：实现UuserDao接口，对数据库操作进行事务管理和错误处理

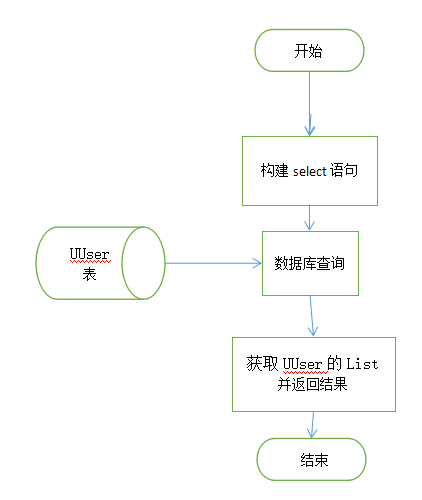
作用：实现对系统用户信息表(UUser)的增删改查

方法说明：

1.List<Uuser> findAll();  
功能：查询所有TapUser

参数：无

返回值：Uuser类的list集合 失败则为null



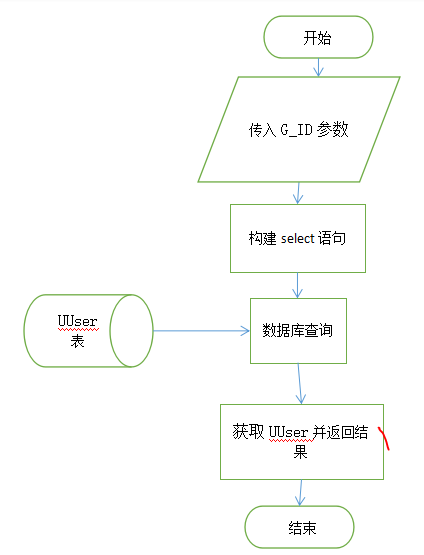
findAll流程图

Uuser findUserById(Integer UU\_ID);

功能：根据id查询TapUser

参数： UU\_ID（用户ID）

返回值：Uuser类 失败则为null



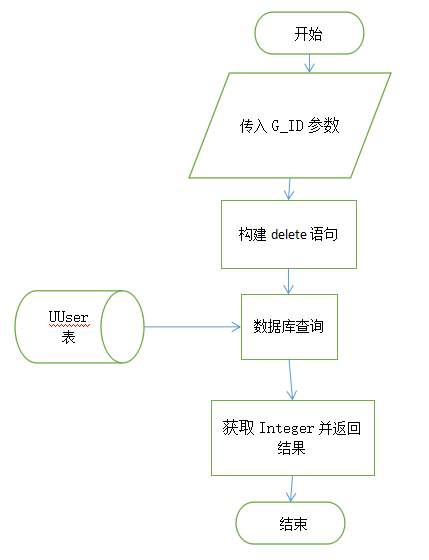
findTapUserById流程图

3.Integer deleteUser(Integer UU\_ID);

功能：根据id删除TapUser

参数： UU\_ID（用户ID）

返回值：Integer成功为1，失败为-1



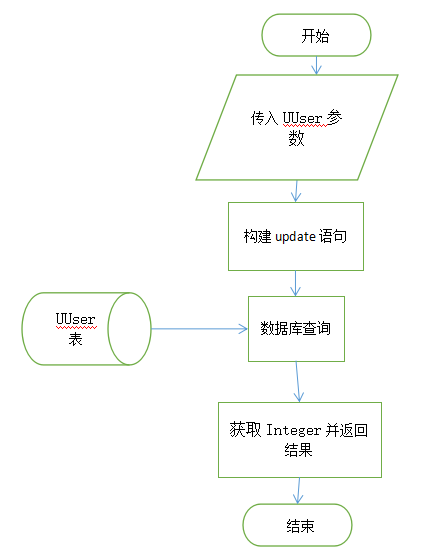
deleteUser流程图

4.Integer updateUser(Uuser uuser);

功能：更新TapUser

参数： Uuser类

返回值：Integer成功为1，失败为-1



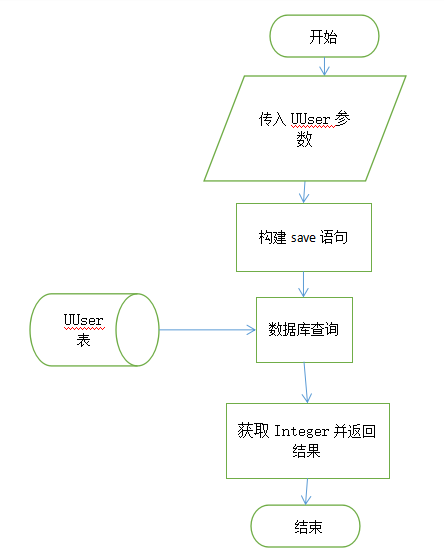
updateUser流程图

5.Integer saveUser(Uuser uuser);

功能：插入一条TapUser

参数： Uuser类

返回值：Integer成功为1，失败为-1



SaveUser流程图

游戏每日评分缓存（GameDayInfoCache）

类说明：实现GameDayInfoCacheDao接口，对数据库操作进行事务管理和错误处理

作用：实现对游戏每日评分缓存表(GameDayInfoCache)的增删改查

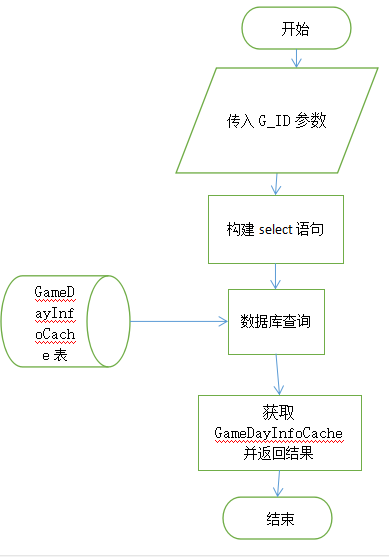
方法说明：

1.public GameDayInfoCache findGameDataCacheById(Integer G\_ID);

功能：根据G\_ID读取缓存数据

参数：G\_ID (游戏id)

返回值：GameDayInfoCache 类 失败为null



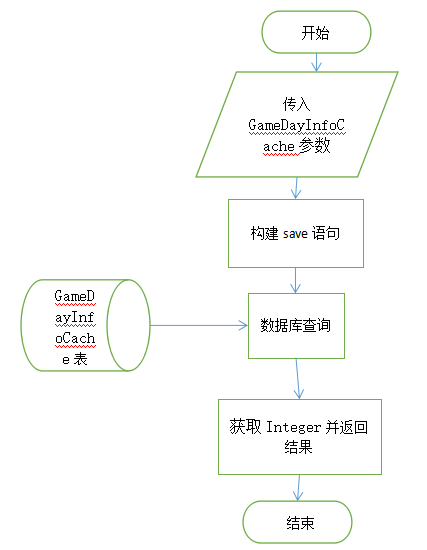
findGameDataCacheById流程图

2.public Integer saveGameDataCache(GameDayInfoCache gameDayInfoCache);

功能：插入缓存数据

参数：GameDayInfoCache类

返回值：Integer 成功为1，失败为-1



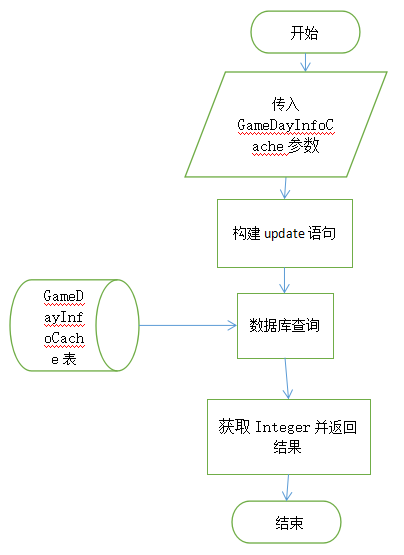
saveGameDataCache流程图

3.public Integer updateGameDataCache(GameDayInfoCache gameDayInfoCache)

功能：更新缓存数据

参数：GameDayInfoCache类

返回值：Integer 成功为1，失败为-1



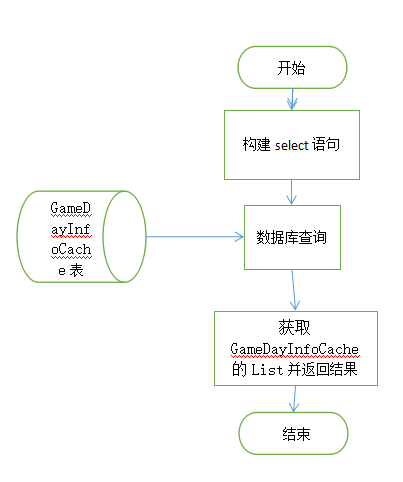
updateGameDataCache流程图

4.public List<T\_GameDayInfoCacheDTO> findAllGameDataCache();

功能：查找所有的id及创建时间

参数：GameDayInfoCache类

返回值：T\_GameDayInfoCacheDTO的List失败为null



findAllGameDataCache流程图

关键词缓存表（KeyWordCache）

KeyWordCacheDaoImpl类

类说明：实现KeyWordCacheDao接口，对数据库操作进行事务管理和错误处理

作用：实现对关键词缓存表(KeyWordCache)的增删改查

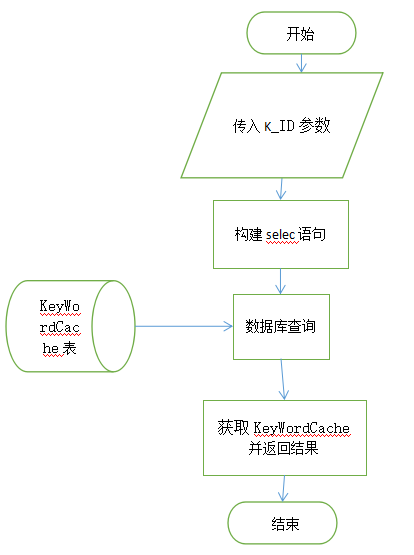
方法说明：

1.public KeyWordCache findKeyWordCacheById(Integer K\_ID);

功能：根据K\_ID读取缓存数据

参数：K\_ID (关键词id)

返回值：KeyWordCache 类 失败为null



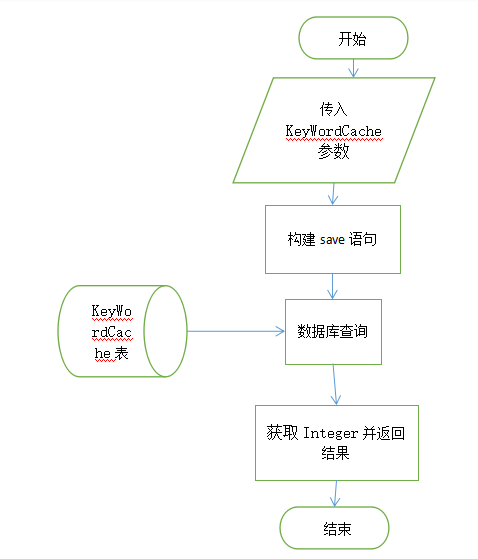
findKeyWordCacheById流程图

2.public Integer saveKeyWordDataCache(KeyWordCache keyWordCache);

功能：插入缓存数据

参数：KeyWordCache类

返回值：Integer 成功为1，失败为-1



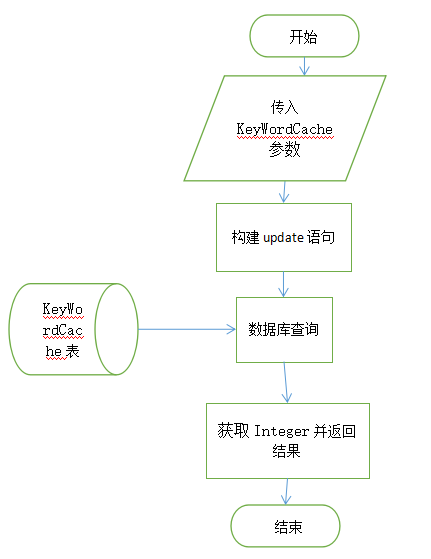
saveKeyWordDataCache流程图

3.public Integer updateKeyWordDataCache(KeyWordCache keyWordCache);

功能：更新缓存数据

参数：KeyWordCache类

返回值：Integer 成功为1，失败为-1



updateKeyWordDataCache流程图

4.public List<KeyWordCache> findAllKeyWordDataCache();

功能：查找所有的id及创建时间

参数：KeyWordCache类

返回值：KeyWordCache的List失败为null