**IoT를 이용한**

**야구장 관리 시스템**

**BS Stadium**

이 주 형 임 연 우

류 명 훈 윤 성 식

구 본 택 김 은 지

강 병 선 노 명 균

**-Client Aspect**

1) 회원 등록/수정/탈퇴

|  |  |  |
| --- | --- | --- |
| **메쏘드명** | **SQL** | **설명** |
| 회원존재여부 | select count(\*) cnt from member where member\_id=?; |  |
| 회원등록 | insert into member values(member\_id,member\_name,member\_password,member\_email,member\_phone,authority); |  |
| 회원정보 수정 | update member set member\_password=?,member\_email=?,member\_phone=? where member\_id=?; |  |
| 회정정보 삭제(탈퇴) | delete from member where member\_id=?; |  |
| 아이디로 회원 찾기 | select \* from member where member\_id=?; |  |

2) 좌석 예매확인 및 취소

|  |  |  |
| --- | --- | --- |
| **메쏘드명** | **SQL** | **설명** |
| 아이디로 구매한 주문번호 검색 | select order\_no from ticket\_order where member\_id=?; |  |
| 회원 예매조회(기간별) | select  t.ticket\_no,t.order\_no,t.game\_no,t.seat\_no,t.member\_gno,t.seat\_fee,t.seat\_fee\_name,o.order\_time,o.member\_id  from ticket t,ticket\_order o where t.order\_no(+)=o.order\_no and o.member\_id=?  and o.order\_time>=sysdate-? and o.order\_time=<sysdate order by o.order\_time desc; | sysdate-?에 7,15,30,60,90 등 기간을 입력한다. |
| 회원 예매조회(일자별(월별)) | select  t.ticket\_no,t.order\_no,t.game\_no,t.seat\_no,t.member\_gno,t.seat\_fee,t.seat\_fee\_name,o.order\_time,o.member\_id  from ticket t,ticket\_order o where t.order\_no(+)=o.order\_no and o.member\_id=?  and to\_char(o.order\_time,'yyyy')=? and to\_char(o.order\_time,'mm')=? order by order\_time desc; | ‘yyyy’에 연도를 입력하고 ‘mm’에 달을 입력한다. |
| 예매취소(티켓 삭제) | delete from ticket\_order where order\_no=?;  delete from ticket where order\_no=?; |  |

**-Manager Aspect**

1) 회원 관리(회원 정보 수정/삭제)

|  |  |  |
| --- | --- | --- |
| **메쏘드명** | **SQL** | **설명** |
| 회원존재여부 | select count(\*) cnt from member where member\_id=?; |  |
| 회원정보 수정 | update member set member\_password=?,member\_email=?,member\_phone=? where member\_id=?; |  |
| 회정정보 삭제(탈퇴) | delete from member where member\_id=?; |  |
| 아이디로 회원 찾기 | select \* from member where member\_id=?; |  |

2) 좌석 관리

|  |  |  |
| --- | --- | --- |
| **메쏘드명** | **SQL** | **설명** |
| updateResvRatio(String resv\_way, int resv\_ratio) | update reservation\_ratio  set resv\_ratio = ?  where resv\_way = ? | 예매비율 설정 |
| updateSeat\_type(int seat\_no, int seat\_type) | update seat set seat\_type = ? where seat\_no = ? | 시즌권 좌석여부(SEAT\_TYPE) 설정  0: 일반좌석,  1: 시즌권좌석 |
| findBySeasonSeats() | select seat\_no, seat.seat\_grade as seat\_grade, seat\_block, seat\_row, seat\_column, seat\_type,  grade\_name  from seat natural join seat\_grade\_name  where seat\_type = 1 | 시즌권 좌석들을 찾기 |
| countSeasonSeats | select count(\*) as count\_season\_seats  from seat  where seat\_type=1; | 시즌권 총 좌석수 |
| updateSeatFee(String seat\_fee\_name, int seat\_fee) | update seat\_fee  set seat\_fee=?  where seat\_fee\_name=? | 좌석가격설정(UPDATE) |
| CreateSeatFee(SeatFeeDto newSeatFeeDto) | insert into seat\_fee (seat\_fee\_name, seat\_grade, member\_gno, seat\_fee)  values (?, ?, ?, ?) | 좌석가격 추가(CREATE) |
| updateSeasonSeatsDiscount(int discount\_rate) | update discount  set discount\_rate=?  where member\_gno=400 | (시즌권좌석) 할인율 설정  MEMBER\_GNO : DISCOUNTNAME  100:성인(일반), 200:어린이(무효데이터)  300:장애인, 400:시즌권좌석 |
| countAllSeats() | select count(\*) as count\_all\_seats from seat | 총 좌석수 |
| countAllReservedSeats(int game\_no) | select count(\*) as count\_all\_reserved\_seats  from ticket  where game\_no=? | 예매된 총 좌석수 |
| countSeatsByGrade(char seat\_grade) | select count(\*) as count\_seats\_by\_grade  from seat  where seat\_grade = ? | 특정등급의 총 좌석수 |
| countReservedSeatsByGrade(char seat\_grade, int game\_no) | select count(\*) as count\_reserved\_seats\_by\_grade  from ticket  join seat  on ticket.seat\_no = seat.seat\_no  where ticket.game\_no=? and seat.seat\_grade = ? | 특정등급의 총 예매된 좌석수 |
| countSeatsByBlock(int seat\_block) | select count(\*) as count\_seats\_by\_block  from seat  where seat\_block=? | 특정블럭의 총 좌석수 |
| countReservedSeatsBy(int seat\_block, int game\_no) | select count(\*) as count\_reserved\_seats\_by\_block  from ticket  join seat  on ticket.seat\_no = seat.seat\_no  where ticket.game\_no=? and seat.seat\_block = ? | 특정블럭의 총 예매된 좌석수 |

3) 매출 관리

4) 전광판 관리

4-1) GAME\_INFO\_DAO

|  |  |  |
| --- | --- | --- |
| **메쏘드명** | **SQL** | **설명** |
| insertGame\_Info(insertGame\_Info: Game\_Info\_Dto): void | insert into game\_info(  game\_no,  inning\_no,  home\_team,  away\_team  )  values (?,?,?,?); | 경기정보를 추가한다. |
| updateGame\_Info(game\_Info: Game\_Info\_Dto): void | update game\_info set  home\_team = ?,  away\_team = ?  where game\_no = ? and inning\_no = ? | 경기번호와 팀 이름에 해당하는 경기정보를 수정한다. |
| deleteGame\_Info(game\_No: int): void | delete game\_info where game\_no = ? and  inning\_no = ?; | 경기정보를 경기번호로 삭제한다. |
| findGame\_Info(game\_No: int): ArrayList<Game\_Info\_Dto> | select \* from game\_info where game\_no = ?; | 경기정보를 경기번호로 검색한다. |

4-2) GAME\_SCHEDULE\_DAO

|  |  |  |
| --- | --- | --- |
| **메쏘드명** | **SQL** | **설명** |
| insertGame\_Schedule(game\_Schedule: Game\_Schedule\_Dto): void | Insert into game\_schedule values(game\_seq.nextval,?,?,?,?) | 경기일정 테이블에 경기일정을 입력한다 |
| updateGame\_Schedule(game\_Schedule: Game\_Schedule\_Dto): void | update game\_schedule set game\_date=?,home\_team=?, away\_team=?, game\_state=? where game\_no=? | 게임번호(pk)에 해당하는 경기일정을 수정한다 |
| deleteGame\_Schedule(game\_No: int): void | delete from game\_schedule where game\_no=? | 게임 번호(pk)에 해당하는 경기일정을 삭제한다 |
| findGame\_Schedule(game\_No: int): Game\_Schedule\_Dto | select \* from game\_schedule where game\_no=? | 게임 번호(pk)에 해당하는 경기일정을 검색한다 |
| findAllGame\_Schedule(): ArrayList<Game\_Schedule\_Dto> | Select \* from game\_schedule | 경기일정 모두를 검색한다 |

4-3) AD\_DAO

|  |  |  |
| --- | --- | --- |
| **메쏘드명** | **SQL** | **설명** |
| insertAd(ad\_No: Ad\_Dto): void | insert into ad values(ad\_seq.nextval,?,?,?) | 광고 테이블에 광고를 추가한다. |
| updateAd(ad: Ad\_Dto): void | update ad set ad\_name=?, ad\_url=?, ad\_playtime=? where ad\_no=? | 광고 번호(pk)에 해당하는 광고정보를 수정한다. |
| deleteAd(ad\_No: int): void | delete from ad where ad\_no = ? | 광고 번호(pk)에 해당하는 광고정보를 삭제한다. |
| findAd(ad\_No:int): Ad\_Dto | select ad\_name, ad\_url, ad\_playtime from ad where ad\_no=? | 광고번호(pk)로 광고정보를 검색한다. |
| findAllAd(): ArrayList<Ad\_Dto> | select ad\_name from ad | 광고이름을 전부 검색한다. |

4-4) TEAM\_DAO

|  |  |  |
| --- | --- | --- |
| **메쏘드명** | **SQL** | **설명** |
| insertTeam(team: Team\_Dto): void | insert into team values(?,?,?) | 팀 테이블에 팀 정보를 입력한다 |
| updateTeam(team: Team\_Dto): void | update team set team\_emblem=?, team\_ishome=? where team\_name=? | 팀 이름(pk)에 해당하는 팀의 정보를 수정한다 |
| deleteTeam(team\_Name: String): void | delete from team where team\_name=? | 팀 이름(pk)에 해당하는 팀을 삭제한다 |
| findTeam(team\_Name: String): Team\_Dto | select \* from team where team\_name=? | 팀 이름(pk)에 해당하는 팀을 검색한다 |
| findAllTeam(): ArrayList<Team\_Dto> | select \* from team | 팀을 모두 검색한다 |

5) CCTV관리(SQL 없음)

**-Client & Manager Common**

1) TICKET\_DAO

|  |  |  |
| --- | --- | --- |
| **메쏘드명** | **SQL** | **설명** |
| insertTicket(ticket: Ticket\_Dto): void | INSERT INTO TICKET VALUES(TICKET\_SEQ.NEXTVAL,?,?,?,?,?,?); | 결제가 완료된 후 티켓 발행할 때 입력 |
| deleteTicket(ticket\_no: int): void | DELETE FROM TICKET  WHERE ORDER\_NO= ?; | 예매가 취소되었을 때 티켓정보 삭제 |
| findTicket(ticket\_no: int): void | SELECT \* FROM TICKET  WHERE TICKET\_NO = ?; | 티켓 번호로 티켓 찾기 |
| findTicketAll(): ArrayList<Ticket\_Dto> | SELECT \* FROM TICKET; | 발행된 티켓 전부 찾기 |

2) TICKET\_ORDER\_DAO

|  |  |  |
| --- | --- | --- |
| **메쏘드명** | **SQL** | **설명** |
| insertTicket\_Order(ticket\_order: Ticket\_Order\_Dto): void | INSERT INTO TICKET\_ORDER  VALUES(ORDER\_SEQ.NEXTVAL,SYSDATE,?); | 티켓 주문 정보 입력 |
| deleteTicket\_Order(order\_no: int): void | DELETE FROM TICKET\_ORDER  WHERE ORDER\_NO = ?; | 티켓 주문 취소 시 삭제 |
| findTicket\_Order(order\_no: int): void | SELECT \* FROM TICKET\_ORDER  WHERE ORDER\_NO = ?; | 티켓 주문 번호로 티켓 찾기 |
| findTicket\_OrderAll(): ArrayList<Ticket\_Order\_Dto> | SELECT \* FROM TICKET\_ORDER; | 티켓 주문 정보 모두 찾기 |