

Komitærn/Human Storage Device

Oblig 2 Adam Aske & Sivert Kindberg

Grov design:

Målet med dette prosjektet er å lage datasystemet som et fengsel bruker. Verkern av oss har vært innsatt eller ansatt ved et fengsel, så dette går ut hva vi tror et fengsel trenger. Fengsler har streng sikkerhet og trenger nøye loggføring av systemene sine. Det er mange ting å holde styring på i et fengsel; fra fanger til matvarer.

Vi designer dette utifra hva vi ser for oss datamaskinen som ansatte i “resepsjonen” bruker. Vi da basere funksjonene vi lager basert på valgene som trengs i det daglige for en fengsel-ansatt.

Vi vil holde styr på fanger, ansatte, rom/celler og bevegelser. Forskjellige ansatte og forskjellige fanger kan ha forskjellige sikkerhetsklareringer, så dette vil vi også holde telling på. Det er viktig for de ansatte å vite hvor fanger er til enhver tid, så vi vil også loggføre når og hvor ansatte og fanger går inn og ut av rom. Vi tar utgangspunkt i at fanger og ansatte enten har en unik kode eller nøkkelkort som gir de adgang til forskjellige rom.

Detaljert design:

Tabeller:

Fanger, Ansatte, Rom, Celler, Varer, Bevegelser og Jobber.

Fanger:

FangeNR, Fornavn, Etternavn, Sikkerhetsnivå, InnsettelseDato, LøslatelseDato, Passord/SikkerhetsNøkkel, Personnummer

For å dehumanisere fangene mest mulig, vil de bli referert til som Fangenummeret deres, derfor er FangeNR primærnøkkelen. For å knytte de til virkeligheten tar vi også personnummer deres, men dette er et for langt nummer til å hele tiden referere til fangene som, derfor er dette ikke primærnøkkelen. Sikkerhetsnivå er et tall som indikerer hvem klarering de har til å bevege seg på innsiden av fengslet. Løslatelse dato gir systemet gi advarsel når fanger er nære løslatelsen sin. Passordet er deres unike nøkkel til å komme inn og ut av fasilitetene.

Ansatte:

AnsattNR, Fornavn, Etternavn, Passord/Sikkerhetsnøkkel, Sikkerhetsnivå

Rom:

RomNR, Sikkerhetsnivå, RomNavn

Rom er alle romene som er ubebode, så ikke en celle/soverommet til en fange. Dette kan være alt fra bibliotek til gymsal. RomNR er et tilfeldig nummer, som er primærnøkkel. Vi bruker et tilfeldig tall istedenfor navnet på rommet som primær nøkkel fordi det kan være flere av samme type rom. Sikkerhetsnivået indikerer hvor høyt

BevegelseLogg:

RomNR, FangeNR, Bevegelse(Inn/Ut), Dato/Tid

RomNR og FangeNR referer til rom og fanger, Bevegelse er en string som forteller om personen gikk inn eller ut av rommet, også blir tiden på bevegelsen lagret.

Celler:

CelleNR er primærnøkkelen, numrene er basert på rekkefølgen de er plassert i fengselet.

FangeNR er en fremmednøkkel som referer til en fange sitt FangeNR, for å holde styring på hvem fange som er i hver celle. En celle er altså soverommet/der en fange bor.

Selections

1. Velge alle som har vært i et rom

```
void FindAllMovementsOfAPrisoner()
{
    std::string fangeNr;
    std::cout << "What prisoner's movement are you looking for?(By FangeNR)?\n";
    getline(std::cin, fangeNr);
    query = "SELECT * FROM Bevegelser WHERE FangeNR = " + fangeNr + "OR AnsattNR = " + fangeNr;
    q = query.c_str();
    qstate = mysql_query(conn, q);
    if (!qstate) {
        res = mysql_store_result(conn);
        while (row = mysql_fetch_row(res)) {
            printf("RomNR: %s\n", row[0]);
        }
    }
    else {
        std::cout << mysql_error(conn);
    }
}
```

2.

Updates And Inserts

1. Sette inn ny fange

```
void InsertNewPrisoner()
{
    std::string fangeNr, fornavn, etternavn, innsetelsedato, løslatelsedato, passord, sikkerhetsnivå, personnummer;
    std::cout << "FangeNR: ";
    getline(std::cin, fangeNr);
    std::cout << "Fornavn: ";
    getline(std::cin, fornavn);
    std::cout << "Etternavn: ";
    getline(std::cin, etternavn);

    query = "INSERT INTO Fanger VALUES(" + fangeNr + ", '" + fornavn + "', '" + etternavn + "', '2012-5-13', '2022-12-24', '4132', 1, 04069749959)";
    q = query.c_str();
    qstate = mysql_query(conn, q);
    if (!qstate) {
        res = mysql_store_result(conn);
    }
    else {
        std::cout << mysql_error(conn);
    }
}
```

2. Registrer entrance til rom

```

void RegistrerBevegelse()
{
    std::string fangeNr, RomNr;
    std::cout << "What prisoner is entering(By FangeNR or AnsattNR)?\n";
    getline(std::cin, fangeNr);
    std::cout << "What room are you entering(By RomNR)?\n";
    getline(std::cin, RomNr);
    query = "INSERT INTO Bevegelse(" + RomNr + ", NULL, '" + fangeNr + "', 'UT', SYSDATE()); ";

    q = query.c_str();
    qstate = mysql_query(conn, q);
    if (!qstate) {
        std::cout << "\nInsertion sucsefull!\n";
    }
    else {
        std::cout << mysql_error(conn) << std::endl;
    }
}

```

3.Slippe fange fri

4.Endre fange som bor i celle

View

```

2 • DROP VIEW IF exists FangerCeller;
3 • CREATE VIEW FangerCeller
4 AS
5 SELECT
6 CelleNR,
7 FangeNR
8 FROM
9 Celler
10 INNER JOIN
11 Fanger USING (FangeNR);
12
13 • select * from FangerCeller;

```

SP

```
1 • use HumanStorageDevice;
2   DELIMITER //
3 • CREATE procedure FinnLøslatelseDato()
4   Begin
5     SELECT FangeNR, Fornavn, Etternavn, LøslatelseDato FROM Fanger;
6   END //
7
8 • CALL FinnLøslatelseDato;
```