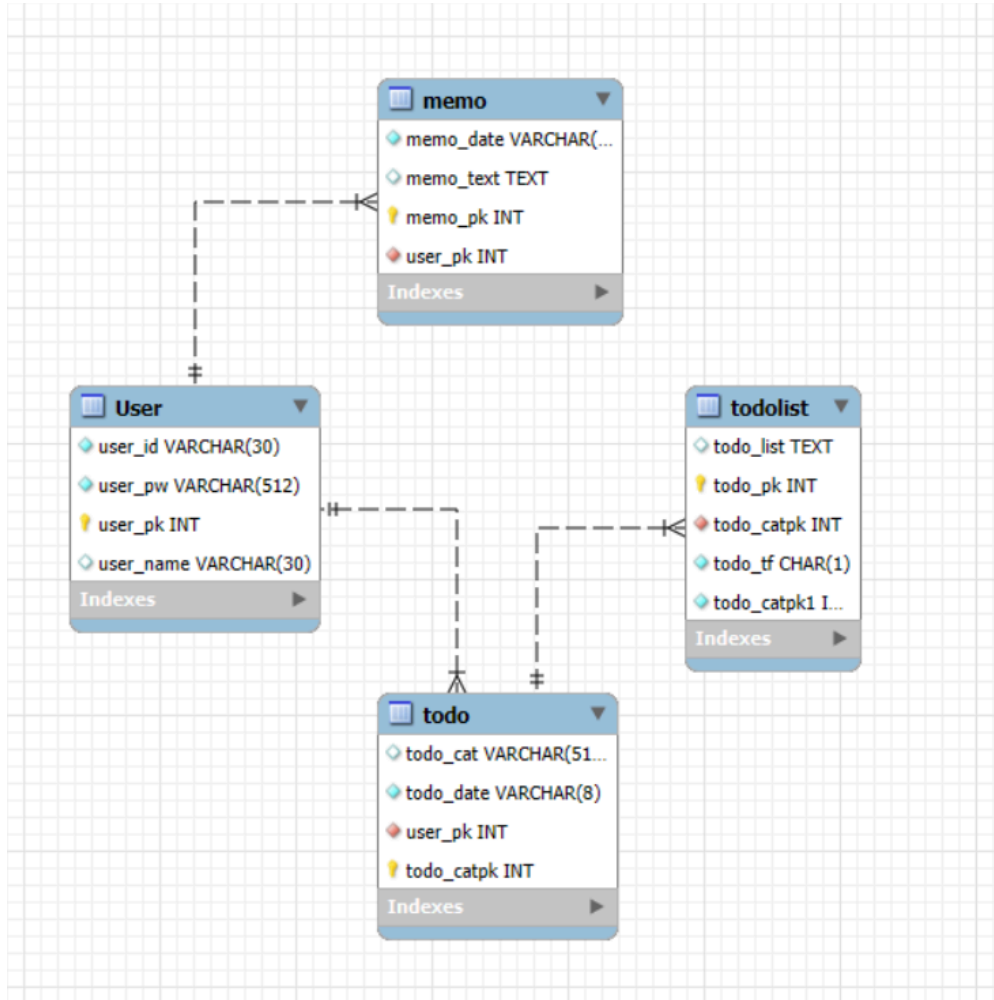




25.04.02



```
get('/dbtest', (req, res) => {
  const sql = "SELECT * from User";
  connection.query(sql, (error, results, fields) => {
    if (error) {
      console.error(error);
      return res.status(500).json({ error: 'DB error' });
    }
    return res.json({ result: results });
  });
});

app.post('/dbtest2', (req, res) => {
  const { pk } = req.body;
  const sql = "SELECT * from User where user_pk = ?;"

  connection.query(sql, pk, (error, results) => {
    return res.status(500).json({result : results});
  });
});
```

```

    })

    })

    //프론트에서 값을 요청하기 위해 보낸 정보를 읽으려면 post를 사용
    //get → (body에 있는 값을 읽을 수 없음 only header만) 값 전송 | post → 프론트에서 보낸 body의 값을 읽고 보낼 수 있다
    //get에서 db에 들어갔을때 user_pk가 7번인 사람의 닉네임, 카테고리, todolist 내용만 뜨게 코딩
    //get → 받아오기 | put → 데이터베이스 내부 값 수정 | post → 삽입

    //여러개 인자 받아서 return하기
    app.post('/dbtest3', (req, res) => {
      const { user_pk, memo_pk } = req.body;
      const sql = "SELECT memo.memo_text from User, memo where User.user_pk = ? and memo.memo_pk = ?;"
      const ex = [user_pk, memo_pk]
      if (!user_pk || !memo_pk) {
        console.log("입력값 오류: ", { user_pk, memo_pk });
        return res.status(502).json({ error: "오류" });
      }
      connection.query(sql, ex, (error, results) => {
        return res.status(500).json({result : results});
      })
    })

    })

    app.listen(port, () => {
      console.log(`Example app listening on port ${port}`)
    })

    //

```

Mysql을 이용하여 서버 생성 및 데이터베이스 테이블 생성 + 백엔드 기초 설명 수강