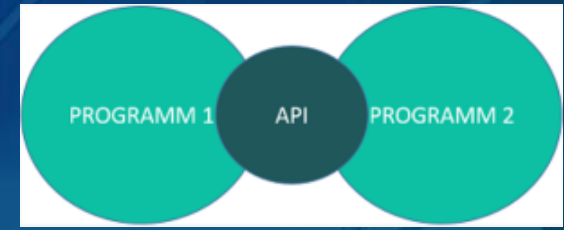
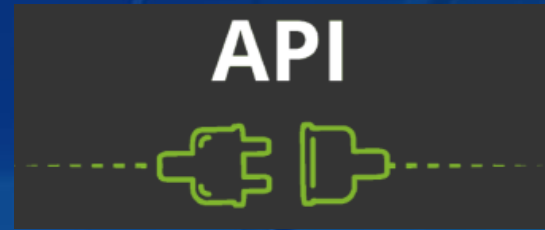


# 웹서버프로그래밍

## 08. 데이터베이스와 SQL, JSP 연동 2





# JDBC (Java Database Connectivity)

클래스와 그 안의 함수(메소드)들의 묶음

JDBC는 Java에서 데이터베이스에 접근 및 활용할 수 있도록 도와주는 API

👉 목적 : 데이터베이스 연결 및 자바 프로그램에서 작성한 'SQL 구문'을 데이터베이스에 전송하여 데이터베이스가 해당 SQL을 처리하도록 하는 것

# JDBC

- JDBC (Java Database Connectivity)

- 데이터베이스는 종류가 다양하기 때문에 개발에 많은 어려움 존재
- Java에서 데이터베이스에 쉽게 접근할 수 있도록 도와주는 인터페이스(API)

☞ 따라서 Java 애플리케이션 개발자는 각 데이터베이스에 대해 자세히 알지 못해도

Java에 기본적으로 포함된 JDBC API만 알면 어떤 데이터베이스에서 동작하는 어플리케이션이든 편하게 개발이 가능

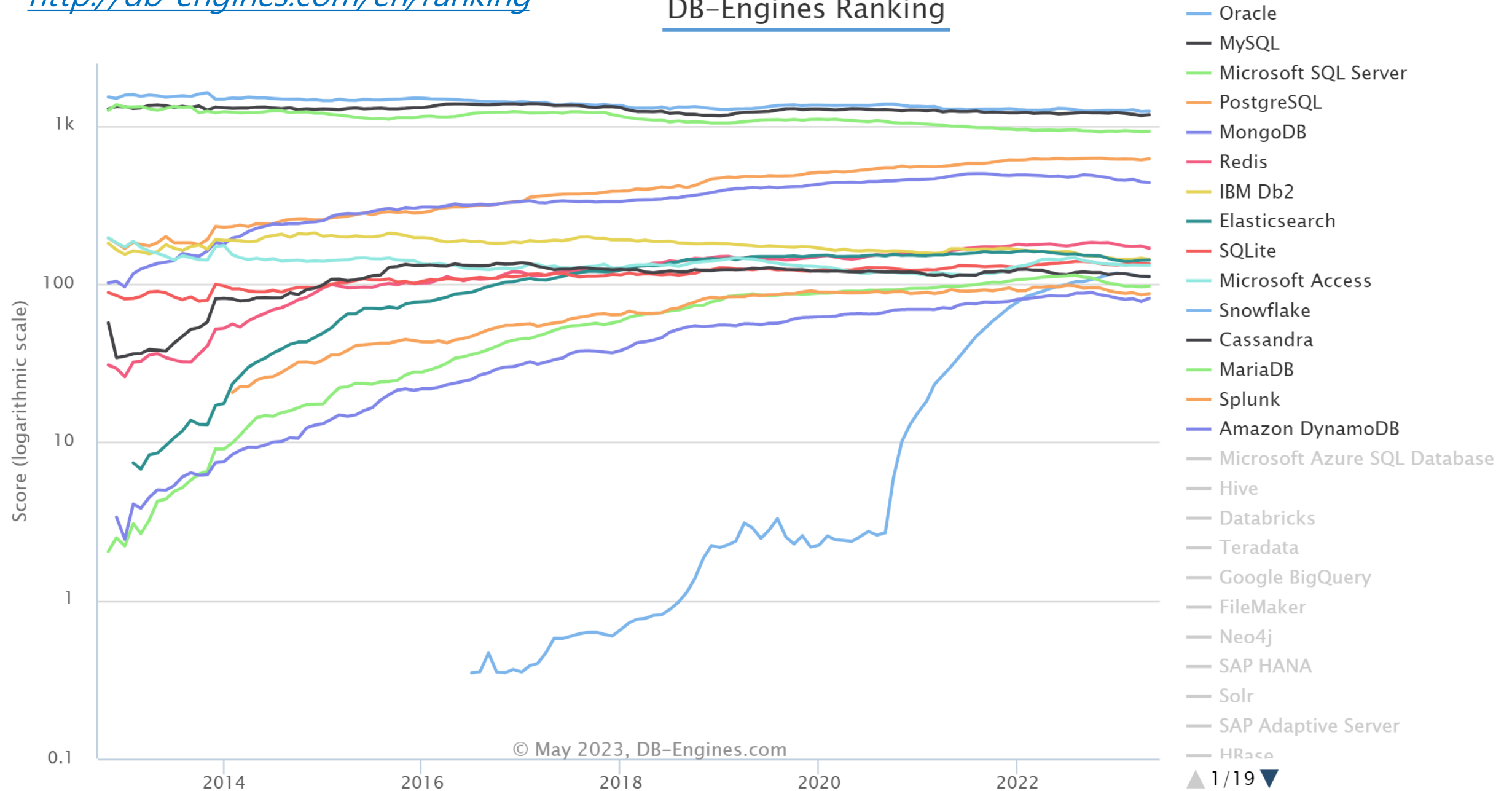
- 즉, JDBC를 이용하면 (자바 프로그래밍을 할 때 ) DBMS의 종류에 관계없이 공통적인 API를 이용하여 데이터베이스 작업을 일관성 있게 처리 가능. 단, DBMS별로 연결을 위한 드라이버 설치가 필요

각 데이터베이스를 만든 곳에서 배포

☞ MySQL을 사용한다면 MySQL 드라이버 필요

(java.sql.Driver 인터페이스 등을) 구현해 놓은 클래스 파일

## DB-Engines Ranking



java.sql.Driver 인터페이스

자바 표준 라이브러리

```
59 public interface Driver {
60
61     /**
62      * Attempts to make a database connection to the given URL.
63      * The driver should return "null" if it realizes it is the wrong kind
64      * of driver to connect to the given URL. This will be common, as when
65      * the JDBC driver manager is asked to connect to a given URL it passes
66      * the URL to each loaded driver in turn.
67      *
68      * :
69      *
70      * Connection connect(String url, java.util.Properties info)
71      *     throws SQLException;
72      *
73      *
74     /**
75      * Retrieves whether the driver thinks that it can open a connection
76      * to the given URL. Typically drivers will return {@code true} if they
77      *
78      * :
79      *
80     int getMajorVersion(); 추상 메소드(멤버 함수)
81
82     /**
83      * Gets the driver's minor version number. Initially this should be 0.
84      * @return this driver's minor version number
85      */
86     int getMinorVersion();
87
88     /**
89      * Reports whether this driver is a genuine JDBC
90      *
91      * :
92      *
93      *
94      *
95      *
96      *
97      *
98      *
99      *
100      *
101      *
102      *
103      *
104      *
105      *
106      *
107      *
108      *
109      *
110      *
111      *
112      *
113      *
114      *
115      *
116      *
117      *
118      *
119      *
120      *
121      *
122      *
123      *
124      *
125      *
126      *
127      *
128      *
129      *
130      *
131      *
132      *
133      *
134      *
135      *
136      *
137      *
138      *
139      *
140      *
141      *
142      *
143      *
144      *
145      *
146      *
147      *
148      *
149      *
150      *
151      *
152      *
153      *
154      *
155      *
156      *
157      *
158      *
159      *
160      *
161      *
162      *
163      *
164      *
165      *
166      *
167      *
168      *
169      *
170      *
171      *
172      *
173      *
174      *
175      *
176      *
177      *
178      *
179      *
180      *
181      *
182      *
183      *
184      *
185      *
186      *
187      *
188      *
189      *
190      *
191      *
192      *
193      *
194      *
195      *
196      *
197      *
198      *
199      *
200      *
201      *
202      *
203      *
204      *
205      *
206      *
207      *
208      *
209      *
210      *
211      *
212      *
213      *
214      *
215      *
216      *
217      *
218      *
219      *
220      *
221      *
222      *
223      *
224      *
225      *
226      *
227      *
228      *
229      *
230      *
231      *
232      *
233      *
234      *
235      *
236      *
237      *
238      *
239      *
240      *
241      *
242      *
243      *
244      *
245      *
246      *
247      *
248      *
249      *
250      *
251      *
252      *
253      *
254      *
255      *
256      *
257      *
258      *
259      *
260      *
261      *
262      *
263      *
264      *
265      *
266      *
267      *
268      *
269      *
270      *
271      *
272      *
273      *
274      *
275      *
276      *
277      *
278      *
279      *
280      *
281      *
282      *
283      *
284      *
285      *
286      *
287      *
288      *
289      *
290      *
291      *
292      *
293      *
294      *
295      *
296      *
297      *
298      *
299      *
300      *
301      *
302      *
303      *
304      *
305      *
306      *
307      *
308      *
309      *
310      *
311      *
312      *
313      *
314      *
315      *
316      *
317      *
318      *
319      *
320      *
321      *
322      *
323      *
324      *
325      *
326      *
327      *
328      *
329      *
330      *
331      *
332      *
333      *
334      *
335      *
336      *
337      *
338      *
339      *
340      *
341      *
342      *
343      *
344      *
345      *
346      *
347      *
348      *
349      *
350      *
351      *
352      *
353      *
354      *
355      *
356      *
357      *
358      *
359      *
360      *
361      *
362      *
363      *
364      *
365      *
366      *
367      *
368      *
369      *
370      *
371      *
372      *
373      *
374      *
375      *
376      *
377      *
378      *
379      *
380      *
381      *
382      *
383      *
384      *
385      *
386      *
387      *
388      *
389      *
390      *
391      *
392      *
393      *
394      *
395      *
396      *
397      *
398      *
399      *
400      *
401      *
402      *
403      *
404      *
405      *
406      *
407      *
408      *
409      *
410      *
411      *
412      *
413      *
414      *
415      *
416      *
417      *
418      *
419      *
420      *
421      *
422      *
423      *
424      *
425      *
426      *
427      *
428      *
429      *
430      *
431      *
432      *
433      *
434      *
435      *
436      *
437      *
438      *
439      *
440      *
441      *
442      *
443      *
444      *
445      *
446      *
447      *
448      *
449      *
450      *
451      *
452      *
453      *
454      *
455      *
456      *
457      *
458      *
459      *
460      *
461      *
462      *
463      *
464      *
465      *
466      *
467      *
468      *
469      *
470      *
471      *
472      *
473      *
474      *
475      *
476      *
477      *
478      *
479      *
480      *
481      *
482      *
483      *
484      *
485      *
486      *
487      *
488      *
489      *
490      *
491      *
492      *
493      *
494      *
495      *
496      *
497      *
498      *
499      *
500      *
501      *
502      *
503      *
504      *
505      *
506      *
507      *
508      *
509      *
510      *
511      *
512      *
513      *
514      *
515      *
516      *
517      *
518      *
519      *
520      *
521      *
522      *
523      *
524      *
525      *
526      *
527      *
528      *
529      *
530      *
531      *
532      *
533      *
534      *
535      *
536      *
537      *
538      *
539      *
540      *
541      *
542      *
543      *
544      *
545      *
546      *
547      *
548      *
549      *
550      *
551      *
552      *
553      *
554      *
555      *
556      *
557      *
558      *
559      *
560      *
561      *
562      *
563      *
564      *
565      *
566      *
567      *
568      *
569      *
570      *
571      *
572      *
573      *
574      *
575      *
576      *
577      *
578      *
579      *
580      *
581      *
582      *
583      *
584      *
585      *
586      *
587      *
588      *
589      *
590      *
591      *
592      *
593      *
594      *
595      *
596      *
597      *
598      *
599      *
600      *
601      *
602      *
603      *
604      *
605      *
606      *
607      *
608      *
609      *
610      *
611      *
612      *
613      *
614      *
615      *
616      *
617      *
618      *
619      *
620      *
621      *
622      *
623      *
624      *
625      *
626      *
627      *
628      *
629      *
630      *
631      *
632      *
633      *
634      *
635      *
636      *
637      *
638      *
639      *
640      *
641      *
642      *
643      *
644      *
645      *
646      *
647      *
648      *
649      *
650      *
651      *
652      *
653      *
654      *
655      *
656      *
657      *
658      *
659      *
660      *
661      *
662      *
663      *
664      *
665      *
666      *
667      *
668      *
669      *
670      *
671      *
672      *
673      *
674      *
675      *
676      *
677      *
678      *
679      *
680      *
681      *
682      *
683      *
684      *
685      *
686      *
687      *
688      *
689      *
690      *
691      *
692      *
693      *
694      *
695      *
696      *
697      *
698      *
699      *
700      *
701      *
702      *
703      *
704      *
705      *
706      *
707      *
708      *
709      *
710      *
711      *
712      *
713      *
714      *
715      *
716      *
717      *
718      *
719      *
720      *
721      *
722      *
723      *
724      *
725      *
726      *
727      *
728      *
729      *
730      *
731      *
732      *
733      *
734      *
735      *
736      *
737      *
738      *
739      *
740      *
741      *
742      *
743      *
744      *
745      *
746      *
747      *
748      *
749      *
750      *
751      *
752      *
753      *
754      *
755      *
756      *
757      *
758      *
759      *
760      *
761      *
762      *
763      *
764      *
765      *
766      *
767      *
768      *
769      *
770      *
771      *
772      *
773      *
774      *
775      *
776      *
777      *
778      *
779      *
780      *
781      *
782      *
783      *
784      *
785      *
786      *
787      *
788      *
789      *
790      *
791      *
792      *
793      *
794      *
795      *
796      *
797      *
798      *
799      *
800      *
801      *
802      *
803      *
804      *
805      *
806      *
807      *
808      *
809      *
810      *
811      *
812      *
813      *
814      *
815      *
816      *
817      *
818      *
819      *
820      *
821      *
822      *
823      *
824      *
825      *
826      *
827      *
828      *
829      *
830      *
831      *
832      *
833      *
834      *
835      *
836      *
837      *
838      *
839      *
840      *
841      *
842      *
843      *
844      *
845      *
846      *
847      *
848      *
849      *
850      *
851      *
852      *
853      *
854      *
855      *
856      *
857      *
858      *
859      *
860      *
861      *
862      *
863      *
864      *
865      *
866      *
867      *
868      *
869      *
870      *
871      *
872      *
873      *
874      *
875      *
876      *
877      *
878      *
879      *
880      *
881      *
882      *
883      *
884      *
885      *
886      *
887      *
888      *
889      *
890      *
891      *
892      *
893      *
894      *
895      *
896      *
897      *
898      *
899      *
900      *
901      *
902      *
903      *
904      *
905      *
906      *
907      *
908      *
909      *
910      *
911      *
912      *
913      *
914      *
915      *
916      *
917      *
918      *
919      *
920      *
921      *
922      *
923      *
924      *
925      *
926      *
927      *
928      *
929      *
930      *
931      *
932      *
933      *
934      *
935      *
936      *
937      *
938      *
939      *
940      *
941      *
942      *
943      *
944      *
945      *
946      *
947      *
948      *
949      *
950      *
951      *
952      *
953      *
954      *
955      *
956      *
957      *
958      *
959      *
960      *
961      *
962      *
963      *
964      *
965      *
966      *
967      *
968      *
969      *
970      *
971      *
972      *
973      *
974      *
975      *
976      *
977      *
978      *
979      *
980      *
981      *
982      *
983      *
984      *
985      *
986      *
987      *
988      *
989      *
990      *
991      *
992      *
993      *
994      *
995      *
996      *
997      *
998      *
999      *
1000      *
1001      *
1002      *
1003      *
1004      *
1005      *
1006      *
1007      *
1008      *
1009      *
1010      *
1011      *
1012      *
1013      *
1014      *
1015      *
1016      *
1017      *
1018      *
1019      *
1020      *
1021      *
1022      *
1023      *
1024      *
1025      *
1026      *
1027      *
1028      *
1029      *
1030      *
1031      *
1032      *
1033      *
1034      *
1035      *
1036      *
1037      *
1038      *
1039      *
1040      *
1041      *
1042      *
1043      *
1044      *
1045      *
1046      *
1047      *
1048      *
1049      *
1050      *
1051      *
1052      *
1053      *
1054      *
1055      *
1056      *
1057      *
1058      *
1059      *
1060      *
1061      *
1062      *
1063      *
1064      *
1065      *
1066      *
1067      *
1068      *
1069      *
1070      *
1071      *
1072      *
1073      *
1074      *
1075      *
1076      *
1077      *
1078      *
1079      *
1080      *
1081      *
1082      *
1083      *
1084      *
1085      *
1086      *
1087      *
1088      *
1089      *
1090      *
1091      *
1092      *
1093      *
1094      *
1095      *
1096      *
1097      *
1098      *
1099      *
1100      *
1101      *
1102      *
1103      *
1104      *
1105      *
1106      *
1107      *
1108      *
1109      *
1110      *
1111      *
1112      *
1113      *
1114      *
1115      *
1116      *
1117      *
1118      *
1119      *
1120      *
1121      *
1122      *
1123      *
1124      *
1125      *
1126      *
1127      *
1128      *
1129      *
1130      *
1131      *
1132      *
1133      *
1134      *
1135      *
1136      *
1137      *
1138      *
1139      *
1140      *
1141      *
1142      *
1143      *
1144      *
1145      *
1146      *
1147      *
1148      *
1149      *
1150      *
1151      *
1152      *
1153      *
1154      *
1155      *
1156      *
1157      *
1158      *
1159      *
1160      *
1161      *
1162      *
1163      *
1164      *
1165      *
1166      *
1167      *
1168      *
1169      *
1170      *
1171      *
1172      *
1173      *
1174      *
1175      *
1176      *
1177      *
1178      *
1179      *
1180      *
1181      *
1182      *
1183      *
1184      *
1185      *
1186      *
1187      *
1188      *
1189      *
1190      *
1191      *
1192      *
1193      *
1194      *
1195      *
1196      *
1197      *
1198      *
1199      *
1200      *
1201      *
1202      *
1203      *
1204      *
1205      *
1206      *
1207      *
1208      *
1209      *
1210      *
1211      *
1212      *
1213      *
1214      *
1215      *
1216      *
1217      *
1218      *
1219      *
1220      *
1221      *
1222      *
1223      *
1224      *
1225      *
1226      *
1227      *
1228      *
1229      *
1230      *
1231      *
1232      *
1233      *
1234      *
1235      *
1236      *
1237      *
1238      *
1239      *
1240      *
1241      *
1242      *
1243      *
1244      *
1245      *
1246      *
1247      *
1248      *
1249      *
1250      *
1251      *
1252      *
1253      *
1254      *
1255      *
1256      *
1257      *
1258      *
1259      *
1260      *
1261      *
1262      *
1263      *
1264      *
1265      *
1266      *
1267      *
1268      *
1269      *
1270      *
1271      *
1272      *
1273      *
1274      *
1275      *
1276      *
1277      *
1278      *
1279      *
1280      *
1281      *
1282      *
1283      *
1284      *
1285      *
1286      *
1287      *
1288      *
1289      *
1290      *
1291      *
1292      *
1293      *
1294      *
1295      *
1296      *
1297      *
1298      *
1299      *
1300      *
1301      *
1302      *
1303      *
1304      *
1305      *
1306      *
1307      *
1308      *
1309      *
1310      *
1311      *
1312      *
1313      *
1314      *
1315      *
1316      *
1317      *
1318      *
1319      *
1320      *
1321      *
1322      *
1323      *
1324      *
1325      *
1326      *
1327      *
1328      *
1329      *
1330      *
1331      *
1332      *
1333      *
1334      *
1335      *
1336      *
1337      *
1338      *
1339      *
1340      *
1341      *
1342      *
1343      *
1344      *
1345      *
1346      *
1347      *
1348      *
1349      *
1350      *
1351      *
1352      *
1353      *
1354      *
1355      *
1356      *
1357      *
1358      *
1359      *
1360      *
1361      *
1362      *
1363      *
1364      *
1365      *
1366      *
1367      *
1368      *
1369      *
1370      *
1371      *
1372      *
1373      *
1374      *
1375      *
1376      *
1377      *
1378      *
1379      *
1380      *
1381      *
1382      *
1383      *
1384      *
1385      *
1386      *
1387      *
1388      *
1389      *
1390      *
1391      *
1392      *
1393      *
1394      *
1395      *
1396      *
1397      *
1398      *
1399      *
1400      *
1401      *
1402      *
1403      *
1404      *
1405      *
1406      *
1407      *
1408      *
1409      *
1410      *
1411      *
1412      *
1413      *
1414      *
1415      *
1416      *
1417      *
1418      *
1419      *
1420      *
1421      *
1422      *
1423      *
1424      *
1425      *
1426      *
1427      *
1428      *
1429      *
1430      *
1431      *
1432      *
1433      *
1434      *
1435      *
1436      *
1437      *
1438      *
1439      *
1440      *
1441      *
1442      *
1443      *
1444      *
1445      *
1446      *
1447      *
1448      *
1449      *
1450      *
1451      *
1452      *
1453      *
1454      *
1455      *
1456      *
1457      *
1458      *
1459      *
1460      *
1461      *
1462      *
1463      *
1464      *
1465      *
1466      *
1467      *
1468      *
1469      *
1470      *
1471      *
1472      *
1473      *
1474      *
1475      *
1476      *
1477      *
1478      *
1479      *
1480      *
1481      *
1482      *
1483      *
1484      *
1485      *
1486      *
1487      *
1488      *
1489      *
1490      *
1491      *
1492      *
1493      *
1494      *
1495      *
1496      *
1497      *
1498      *
1499      *
1500      *
1501      *
1502      *
1503      *
1504      *
1505      *
1506      *
1507      *
1508      *
1509      *
1510      *
1511      *
1512      *
1513      *
1514      *
1515      *
1516      *
1517      *
1518      *
1519      *
1520      *
1521      *
1522      *
1523      *
1524      *
1525      *
1526      *
1527      *
1528      *
1529      *
1530      *
1531      *
1532      *
1533      *
1534      *
1535      *
1536      *
1537      *
1538      *
1539      *
1540      *
1541      *
1542      *
1543      *
1544      *
1545      *
1546      *
1547      *
1548      *
1549      *
1550      *
1551      *
1552      *
1553      *
1554      *
1555      *
1556      *
1557      *
1558      *
1559      *
1560      *
1561      *
1562      *
1563      *
1564      *
1565      *
1566      *
1567      *
1568      *
1569      *
1570      *
1571      *
1572      *
1573      *
1574      *
1575      *
1576      *
1577      *
1578      *
1579      *
1580      *
1581      *
1582      *
1583      *
1584      *
1585      *
1586      *
1587      *
1588      *
1589      *
1590      *
1591      *
1592      *
1593      *
1594      *
1595      *
1596      *
1597      *
1598      *
1599      *
1600      *
1601      *
1602      *
1603      *
1604      *
1605      *
1606      *
1607      *
1608      *
1609      *
1610      *
1611      *
1612      *
1613      *
1614      *
1615      *
1616      *
1617      *
1618      *
1619      *
1620      *
1621      *
1622      *
1623      *
1624      *
1625      *
1626      *
1627      *
1628      *
1629      *
1630      *
1631      *
1632      *
1633      *
1634      *
1635      *
1636      *
1637      *
1638      *
1639      *
1640      *
1641      *
1642      *
1643      *
1644      *
1645      *
1646      *
1647      *
1648      *
1649      *
1650      *
1651      *
1652      *
1653      *
1654      *
1655      *
1656      *
1657      *
1658      *
1659      *
1660      *
1661      *
1662      *
1663      *
1664      *
1665      *
1666      *
1667      *
1668      *
1669      *
1670      *
1671      *
1672      *
1673      *
1674      *
1675      *
1676      *
1677      *
1678      *
1679      *
1680      *
1681      *
1682      *
1683      *
1684      *
1685      *
1686      *
1687      *
1688      *
1689      *
1690      *
1691      *
1692      *
1693      *
1694      *
1695      *
1696      *
1697      *
1698      *
1699      *
1700      *
1701      *
1702      *
1703      *
1704      *
1705      *
1706      *
1707      *
1708      *
1709      *
1710      *
1711      *
1712      *
1713      *
1714      *
1715      *
1716      *
1717      *
1718      *
1719      *
1720      *
1721      *
1722      *
1723      *
1724      *
1725      *
1726      *
1727      *
1728      *
1729      *
1730      *
1731      *
1732      *
1733      *
1734      *
1735      *
1736      *
1737      *
1738      *
1739      *
1740      *
1741      *
1742      *
1743      *
1744      *
1745      *
1746      *
1747      *
1748      *
1749      *
1750      *
1751      *
1752      *
1753      *
1754      *
1755      *
1756      *
1757      *
1758      *
1759      *
1760      *
1761      *
1762      *
1763      *
1764      *
1765      *
1766      *
1767      *
1768      *
1769      *
1770      *
1771      *
1772      *
1773      *
1774      *
1775      *
1776      *
1777      *
1778      *
1779      *
1780      *
1781      *
1782      *
1783      *
1784      *
1785      *
1786      *
1787      *
1788      *
1789      *
1790      *
1791      *
1792      *
1793      *
1794      *
1795      *
1796      *
1797      *
1798      *
1799      *
1800      *
1801      *
1802      *
1803      *
1804      *
1805      *
1806      *
1807      *
1808      *
1809      *
1810      *
1811      *
1812      *
1813      *
1814      *
1815      *
1816      *
1817      *
1818      *
1819      *
1820      *
1821      *
1822      *
1823      *
1824      *
1825      *
1826      *
1827      *
1828      *
1829      *
1830      *
1831      *
1832      *
1833      *
1834      *
1835      *
1836      *
1837      *
1838      *
1839      *
1840      *
1841      *
1842      *
1843      *
1844      *
1845      *
1846      *
1847      *
1848      *
1849      *
1850      *
1851      *
1852      *
1853      *
1854      *
1855      *
1856      *
1857      *
1858      *
1859      *
1860      *
1861      *
1862      *
1863      *
1864      *
1865      *
1866      *
1867      *
1868      *
1869      *
1870      *
1871      *
1872      *
1873      *
1874      *
1875      *
1876      *
1877      *
1878      *
1879      *
1880      *
1881      *
1882      *
1883      *
1884      *
1885      *
1886      *
1887      *
1888      *
1889      *
1890      *
1891      *
1892      *
1893      *
1894      *
1895      *
1896      *
1897      *
1898      *
1899      *
1900      *
1901      *
1902      *
1903      *
1904      *
1905      *
1906      *
1907      *
1908      *
1909      *
1910      *
1911      *
1912      *
1913      *
1914      *
1915      *
1916      *
1917      *
1918      *
1919      *
1920      *
1921      *
1922      *
1923      *
1924      *
1925      *
1926      *
1927      *
1928      *
1929      *
1930      *
1931      *
1932     
```

# JDBC

- ☞ 우리는 자바에 기본적으로 포함된 JDBC를 사용해 프로그램 코드를 작성하고,
- 실제 데이터베이스 연결은 각 데이터베이스 회사가 제공하는 JDBC 드라이버(JDBC API 구현 클래스)를 이용
- DB 서버에 원격으로 접속하기 위한 네트워크 연결 제공
  - Java 프로그램에서 DB 검색과 삽입, 수정, 삭제 등을 위한 SQL문을 전송하고,  
서버로부터 결과를 받을 수 있는 기능 제공
  - 데이터베이스 연결 및 자바 프로그램에서 작성한 SQL 구문을 데이터베이스에 전송하여  
데이터베이스가 해당 SQL을 처리

## MySQL 드라이버의 <connect 메소드> 구현 예

```
@Override
public Connection connect(String url, Properties info) throws SQLException {
    if (!acceptsURL(url)) {
        return null;
    }

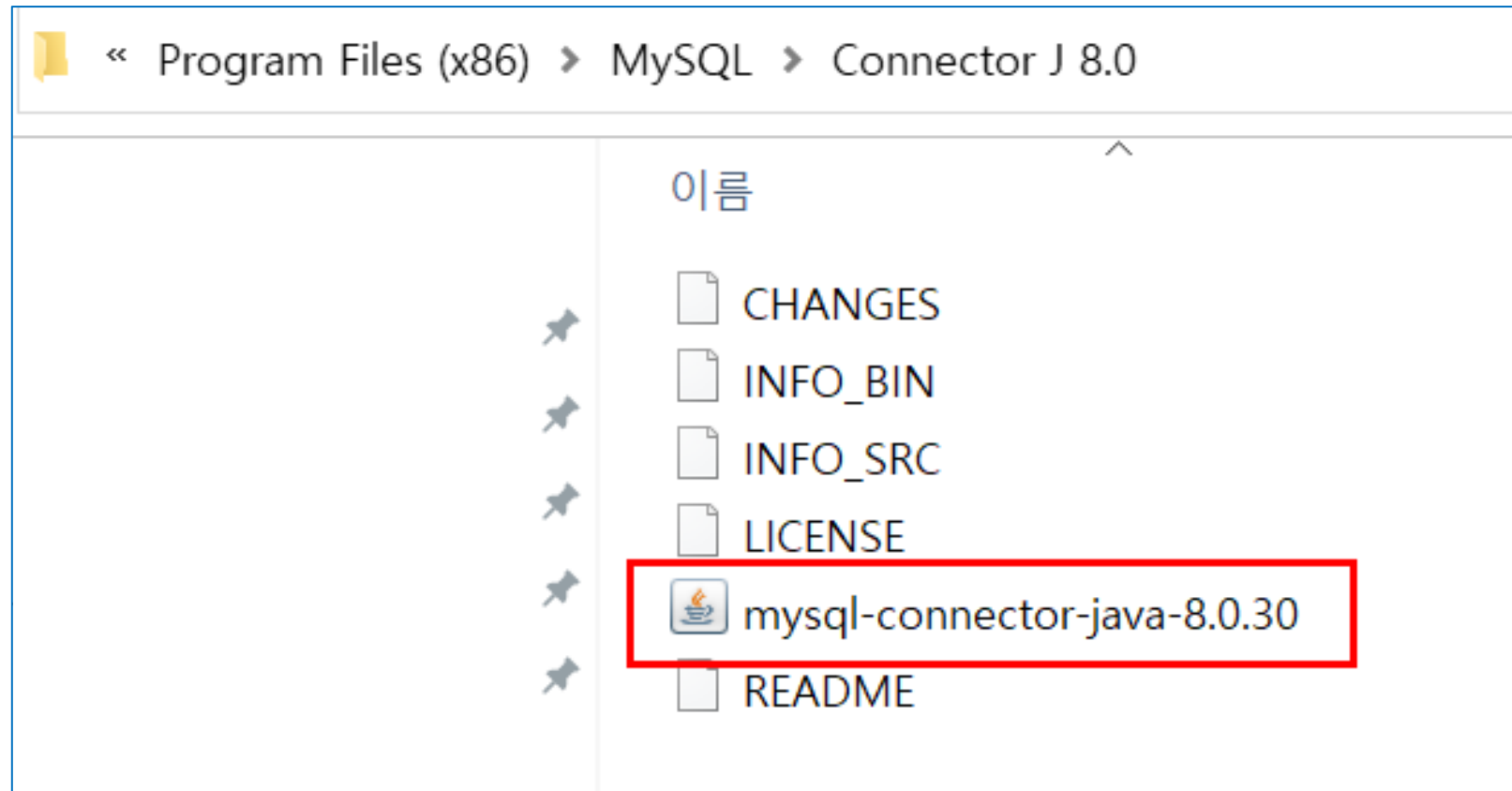
    String host = null;
    int port = -1;
    String database = null;

    Properties props = null;
    if (info != null) {
        props = (Properties) info.clone();
    } else {
        props = new Properties();
    }

    // Parse URL
    if (url.startsWith(JDBC_MYSQL_PREFIX)) {
        url = url.substring(JDBC_MYSQL_PREFIX.length());
    } else if (url.startsWith(JDBC_MYSQL_LEGACY_PREFIX)) {
        url = url.substring(JDBC_MYSQL_LEGACY_PREFIX.length());
        props.setProperty("useLegacyDatetimeCode", "false");
    } else {
        return null;
    }
}
```

# MySQL JDBC 드라이버 설치

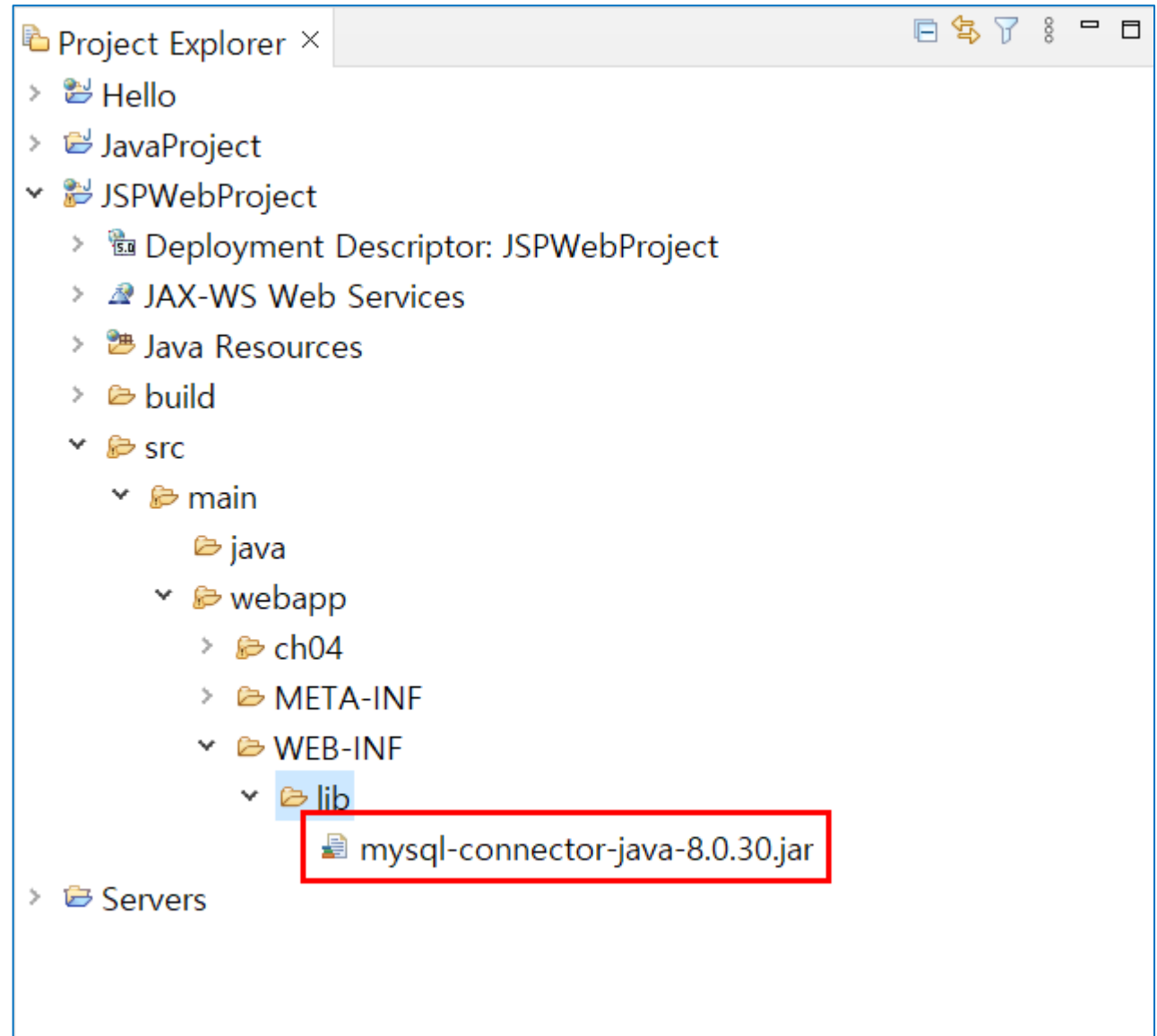
- JDBC 위치 확인, ctrl+c 복사





# MySQL JDBC 드라이버 설치

- 오른쪽 위치에 ctrl+v 해서 붙여넣기



## [참고] MySQL JDBC 드라이버 설치 방법

### 1. 톰켓이 설치된 경로 하위의 lib 폴더에 추가하는 방법

- 한 번의 설정으로 해당 컴퓨터에서 실행되는 모든 웹 어플리케이션에 적용
- 편리해 보일 수 있으나 차후 웹 어플리케이션 배포 파일에 드라이버가 포함되지 않으므로 별도의 설정 필요, 서버가 시작될 때 자동으로 인식

### 2. 개별 프로젝트의 WEB-INF 하위의 lib 폴더에 추가하는 방법

- 프로젝트마다 동일한 설정을 반복해야 하므로 처음에는 번거로우나 작업 공간을 변경하거나 배포 시에도 드라이버가 함께 따라 간다는 편리함 존재

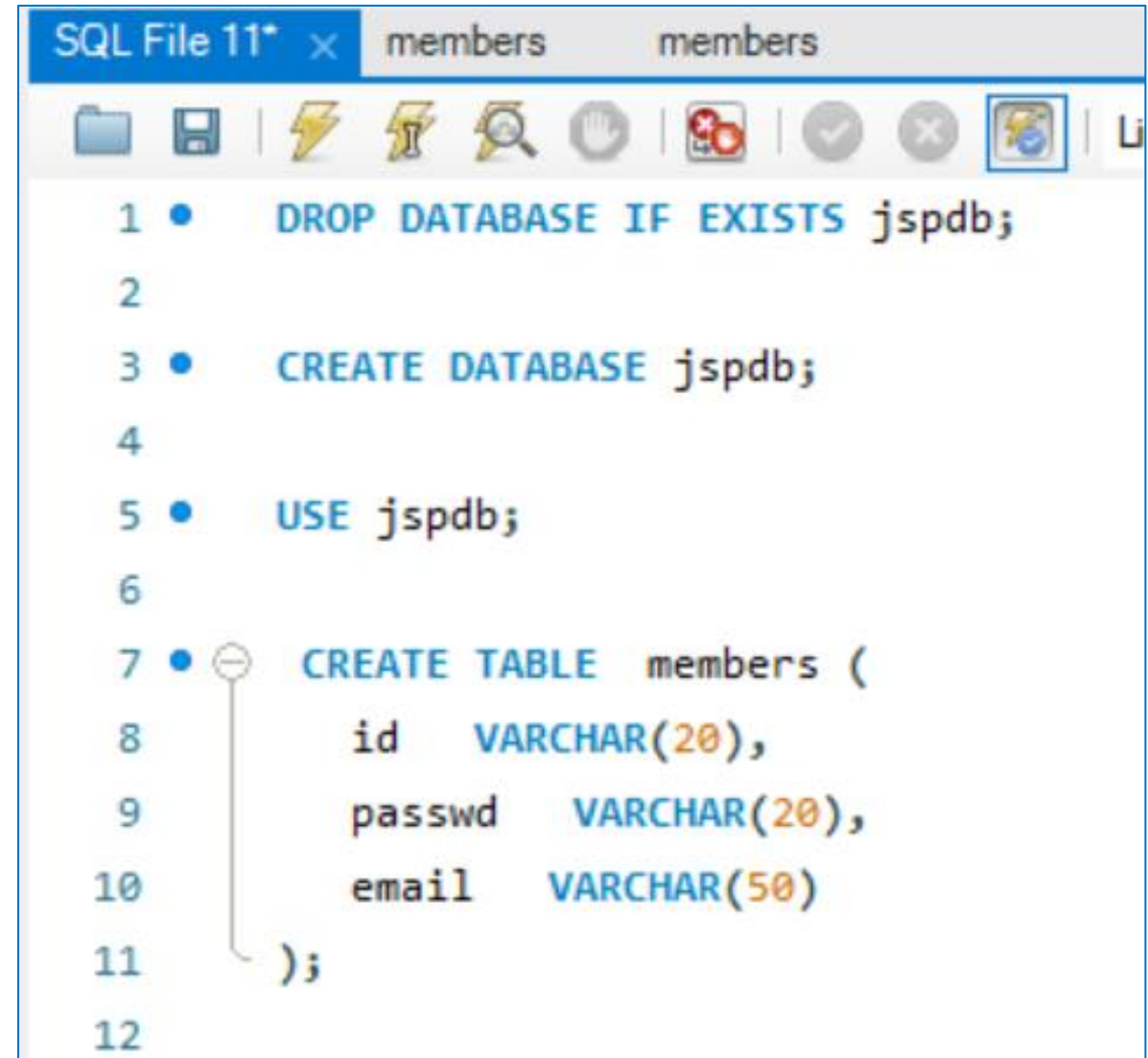
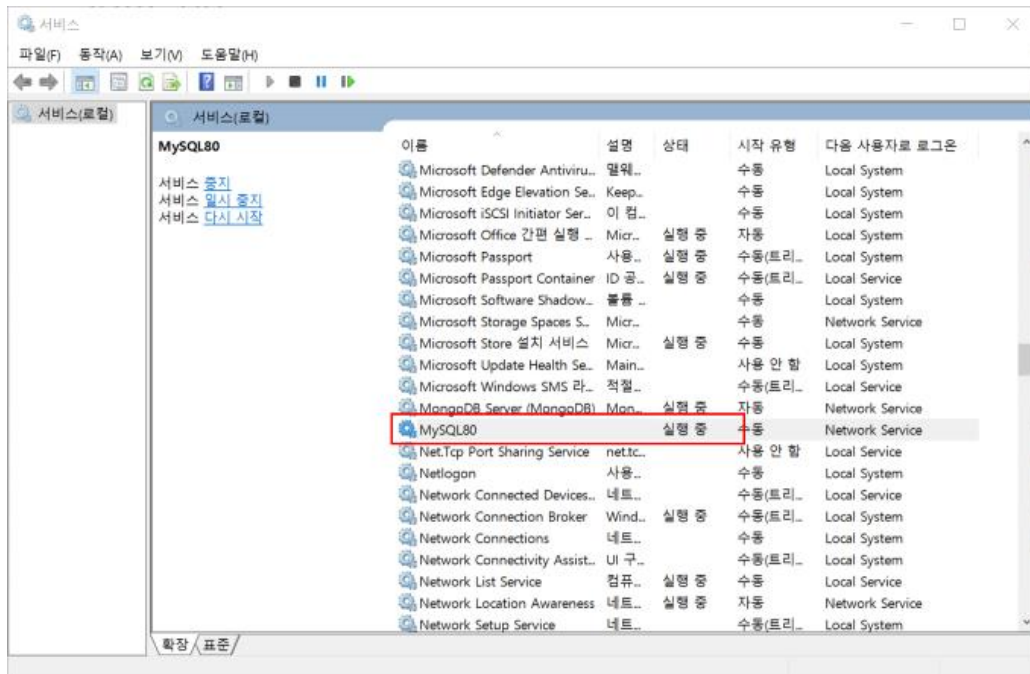
# JDBC 프로그래밍

(JDBC를 이용한) 데이터베이스 연동 및 SQL문을 처리하기 위한 방법에 대해 학습

# JDBC 연동

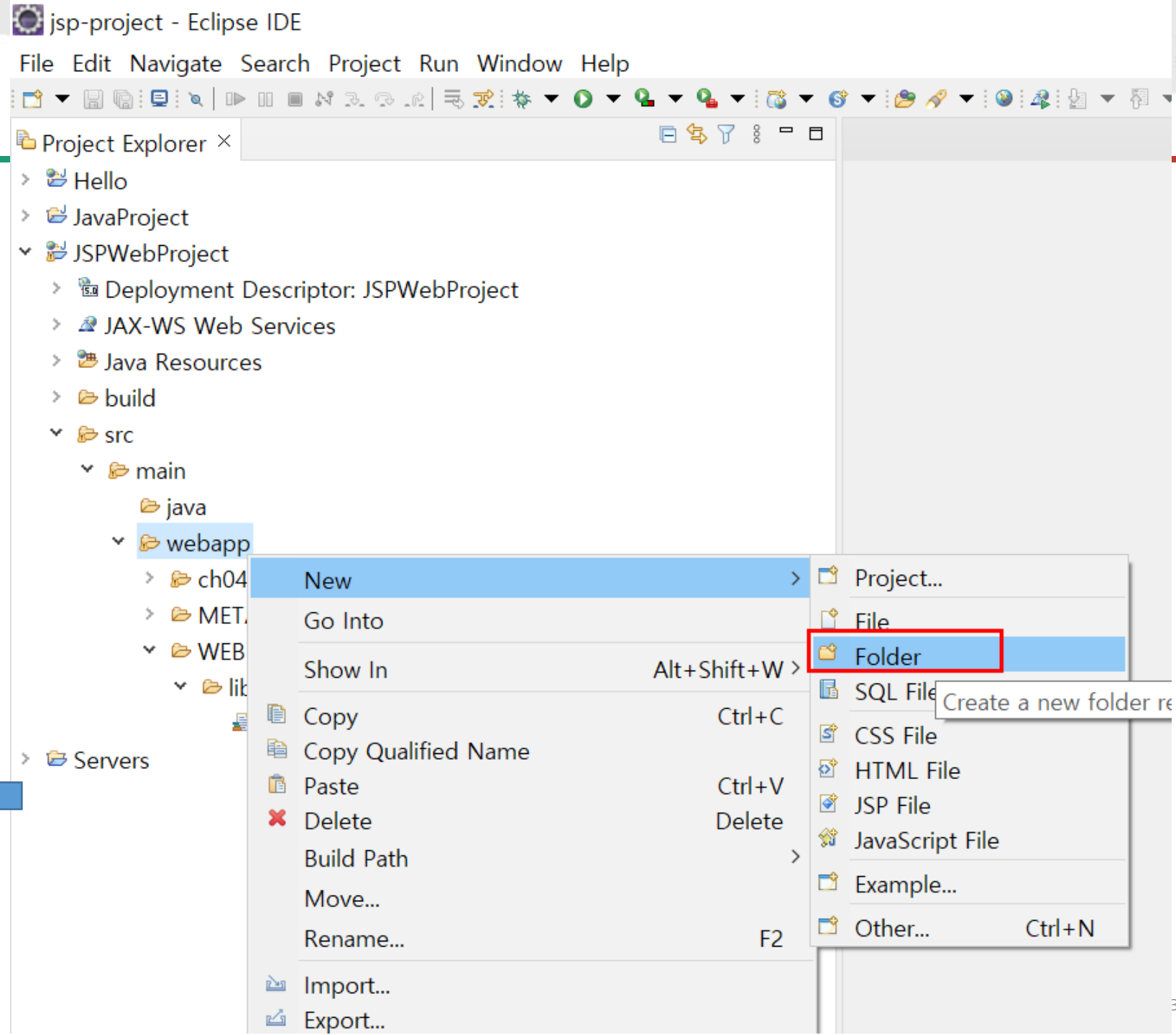
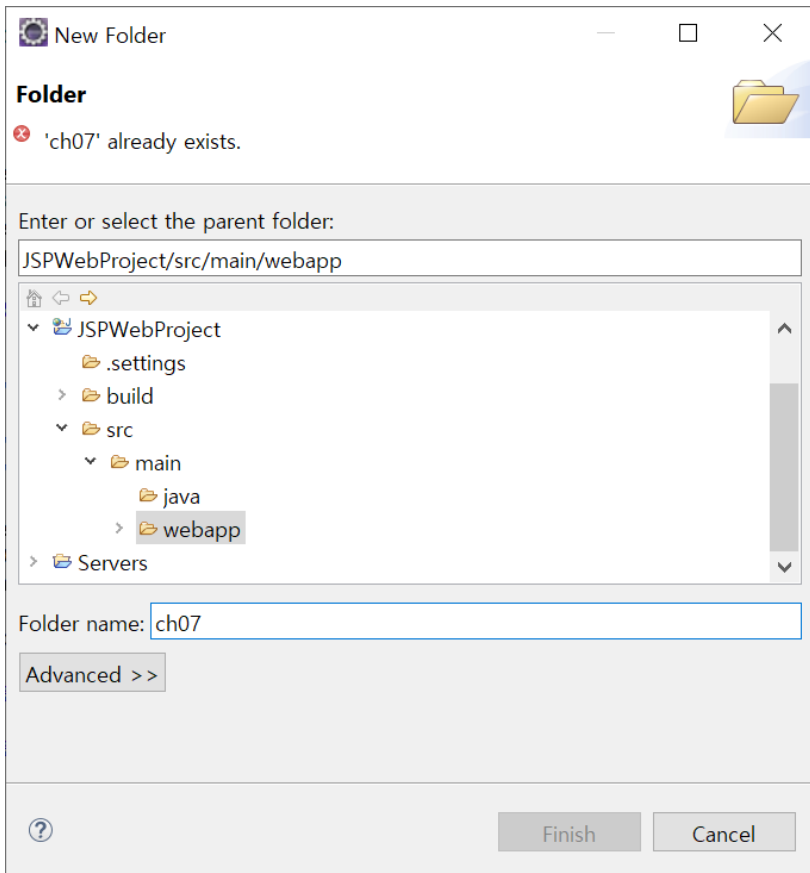
- MySQL에서 DB 생성

※ MySQL 서버 실행 여부 확인



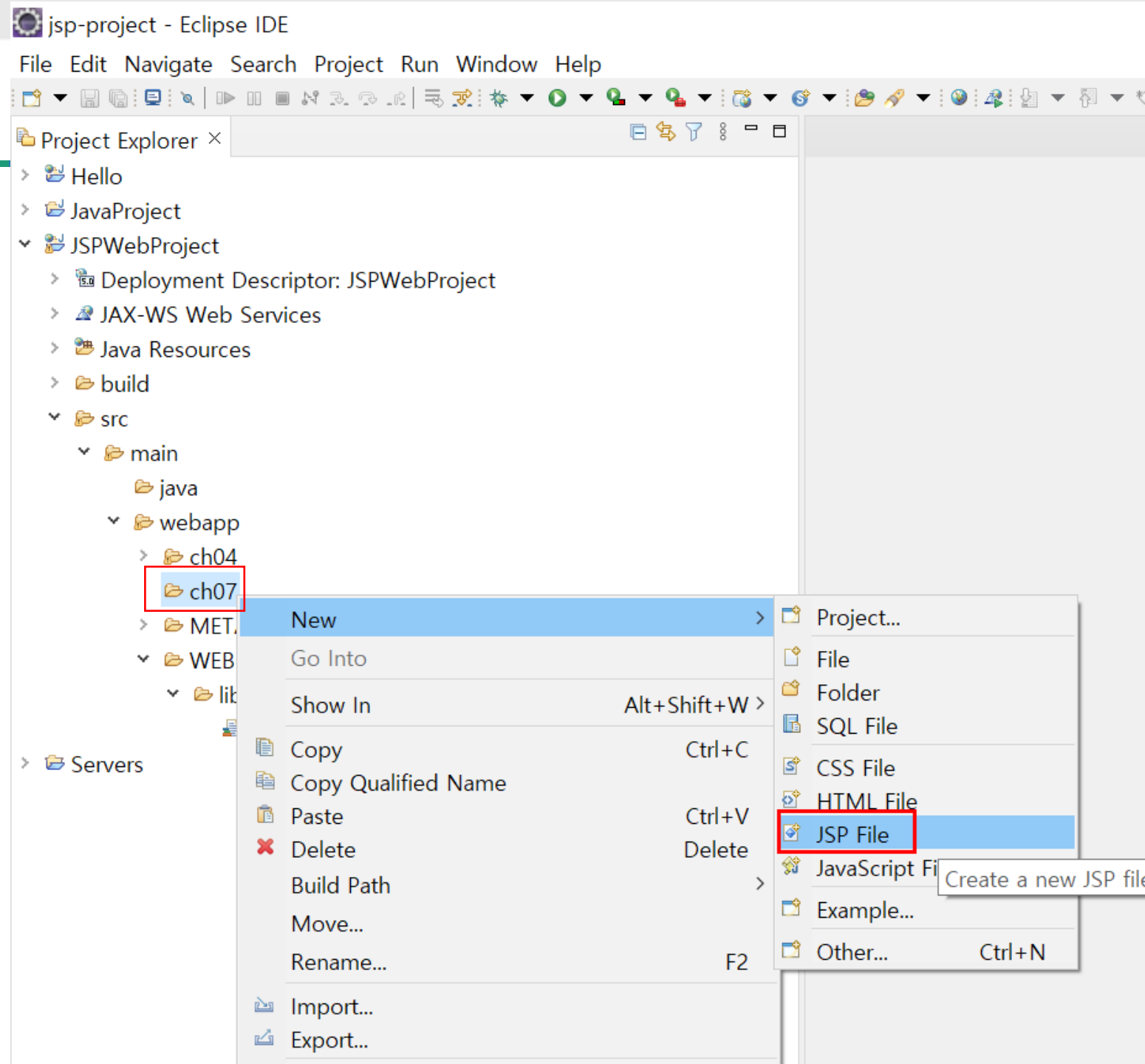
# JDBC 연동

## ■ ch07 폴더 생성



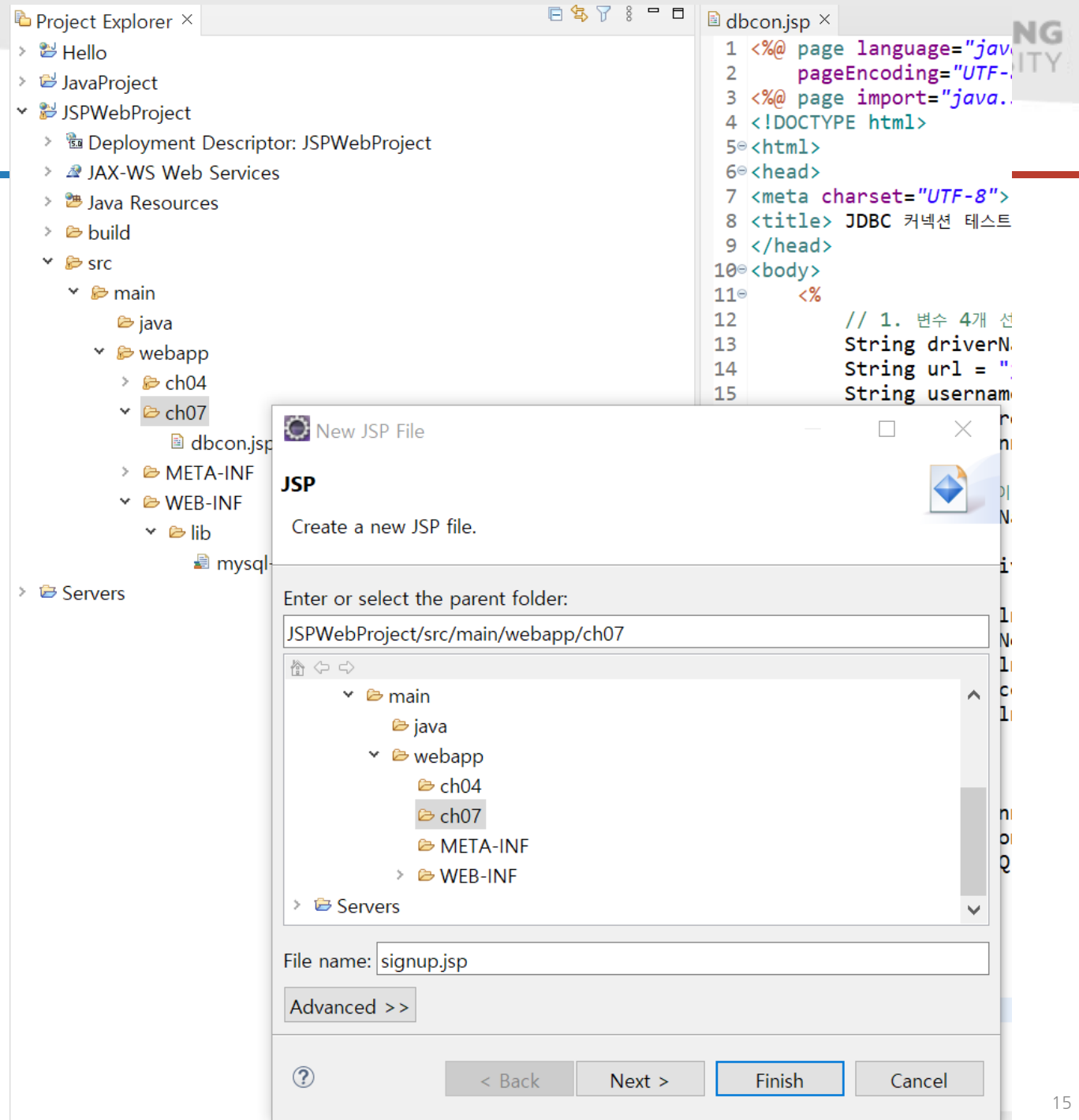
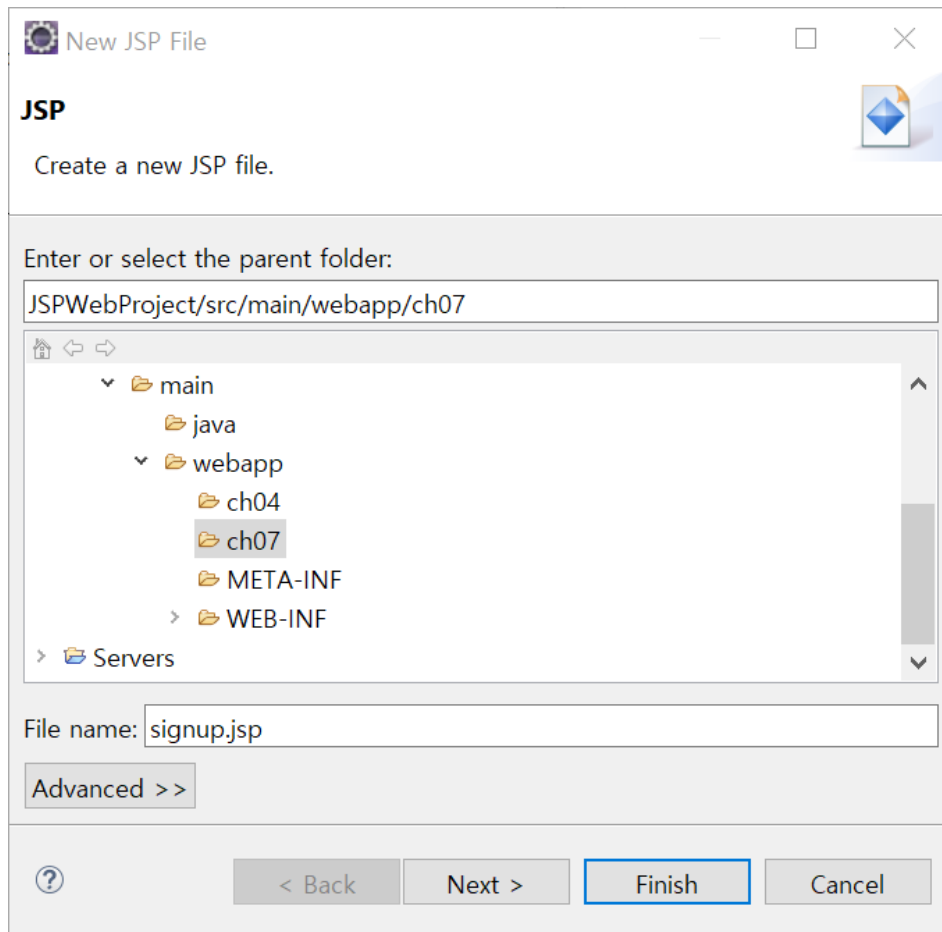
# JDBC 연동

- ch07 폴더에 새로운 jsp 파일 생성



# JDBC 연동

## ■ signup.jsp 파일 생성



# JDBC 연동

## ■ 실행 결과

signup.jsp

Home > Signup the Membership

회원 가입

아이디 :

비밀번호 :

이메일 :

가입하기 다시작성



# JDBC 연동

- signup.jsp 코드 입력

signup.jsp ×

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6     <meta charset="UTF-8">
7     <title> 회원 가입 </title>
8 </head>
9 <body>
10    Home > Signup the Membership
11    <hr>
12    <form action="signup_process.jsp" name="user_info" method="get">
13        <fieldset style="width:200px">
14            <legend> 회원 가입 </legend><p>
15
16                아이디 : <br>
17                <input type="text" name="userID"><br><br>
18
19                비밀번호 : <br>
20                <input type="password" name="userPW"><br><br>
21
22                이메일 : <br>
23                <input type="email" name="userMAIL"><br><br>
24                <hr>
25
26                <div align="center">
27                    <input type="submit" value="가입하기 "> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
28                    <input type="reset" value="다시작성 ">
29                </div><br>
30            </fieldset>
31        </form>
32    </body>
33 </html>
```

# JDBC 연동

## ■ 결과 확인(signup\_process.jsp) signup.jsp

회원 가입

아이디 :  
hansung

비밀번호 :  
.....

이메일 :  
hansung@hansung.ac.kr

가입하기 다시작성

userID = 1234

userPW = hasung

userMAIL = hansung@hansung.ac.kr

## signup\_process.jsp

가입 안내

회원가입 성공!

signup\_process.jsp 파일에서는  
1. jspdb 데이터베이스에 회원정보를 저장한 후  
2. 이와 같은 웹 페이지를 웹브라우저로 전송

☞ 다음과 같은 형태로 데이터가 URL 뒤에 붙어서 서버로 전달

? userID = 1234 & userPW = hasung & userMAIL = hansung@hansung.ac.kr

# JDBC 연동

## ■ signup\_process.jsp 코드 입력

다음과 같은 SQL 문을 만들어서 MySQL 서버로 전송

```
INSERT INTO members
VALUES ('hansung', '1234', 'hansung@hansung.ac.kr');
```

```
signup.jsp *signup_process.jsp ×
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2 pageEncoding="UTF-8"%>
3 <%@ page import="java.sql.*" %>
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <meta charset="UTF-8">
8 <title> 가입 안내 </title>
9 </head>
10 <body>
11
12 <%
13     String u_id = request.getParameter("userID");
14     String u_pw = request.getParameter("userPW");
15     String u_mail = request.getParameter("userMAIL");
16
17     String sql = "INSERT INTO members VALUES";
18     sql += "('' + u_id + "',''' + u_pw + "',''' + u_mail + '')";
19
20     String driverName="com.mysql.jdbc.Driver";
21     Class.forName(driverName);
22
23     String url = "jdbc:mysql://localhost:3306/jspdb";
24     String username = "root";
25     String password = "1234";
26
27     Connection conn = DriverManager.getConnection(url, username, password);
28     Statement sm = conn.createStatement();
29
30     int count = sm.executeUpdate(sql);
31     if(count == 1){
32         out.println("회원가입 성공!");
33     }else{
34         out.println("회원가입 실패!");
35     }
36     sm.close();
37     conn.close();
38 %>
39
40 </body>
41
42 </html>
```

# 설명

- SQL 문을 작성해 주는 Java 코드

```
String sql = "INSERT INTO members VALUES";  
sql += "(" + u_id + "," + u_pw + "," + u_mail + ")";
```

아래 SQL문과 같은 것

```
String sql = "INSERT INTO members VALUES" + "(" + u_id + "," + u_pw + "," + u_mail + ")";
```

만약 사용자가 입력한 데이터가

u\_id = "hasung", u\_pw = "1234", u\_mail = "hansung@hansung.ac.kr" 이라고 한다면

생성되는 SQL문은 다음과 같음

```
INSERT INTO members VALUES ('hasung', '1234', 'hansung@hansung.ac.kr');
```

# INSERT

- 다음 두 SQL 문은 동일한 동작 수행

	id	passwd	email
①	hansung	1234	hansung@hansung.ac.kr
②	hansung	1234	hansung@hansung.ac.kr
③	hansung	NULL	hansung@hansung.ac.kr
⑤	hansung	1234	NULL

① `INSERT INTO members(id, passwd, email) VALUES ('hansung', '1234', 'hansung@hansung.ac.kr');`  
(속성 이름)을 적어주는 것이 가능

② `INSERT INTO members VALUES ('hansung', '1234', 'hansung@hansung.ac.kr');`

- (속성 이름을 적어주었을 경우) 다음과 같이 입력하고 싶은 데이터만 입력 가능

③ `INSERT INTO members(id, email) VALUES ('hansung', 'hansung@hansung.ac.kr');`

- 다음은 오류 발생(테이블 이름만 적을 경우 모든 데이터를 나열해야 함)

④ `INSERT INTO members VALUES ('hansung', '1234');` X (오류)

❌ Error Code: 1136. Column count doesn't match value count at row 1

⑤ `INSERT INTO members VALUES ('hansung', '1234', NULL);` O

# MySQL 드라이버 로딩

# MySQL 드라이버 로딩

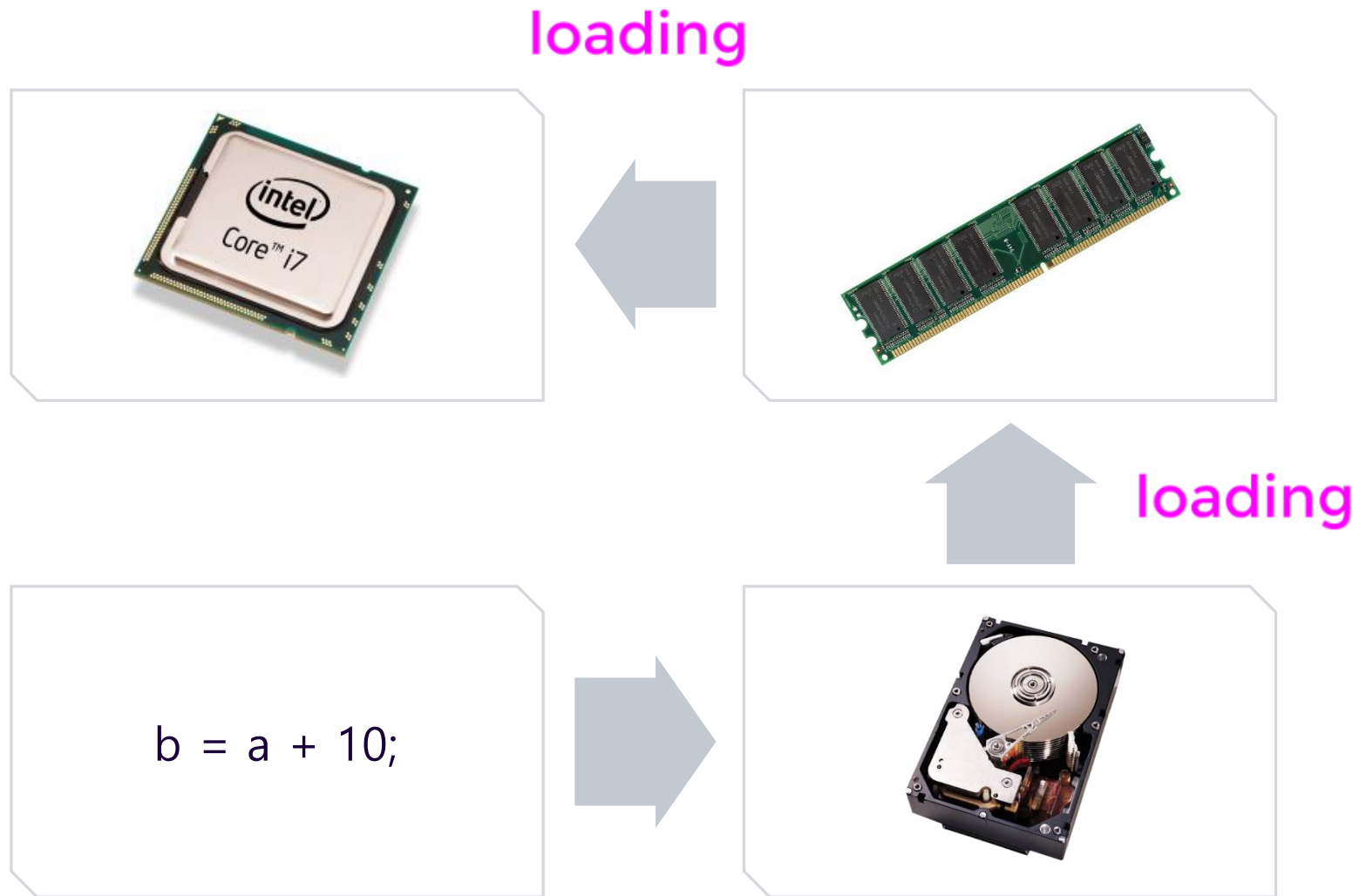
## ■ MySQL 드라이버 클래스 로딩

```
<%  
String u_id = request.getParameter("userID");  
String u_pw = request.getParameter("userPW");  
String u_mail = request.getParameter("userMAIL");  
  
String sql = "INSERT INTO members VALUES";  
sql += "(" + u_id + "',' + u_pw + "',' + u_mail + "'";  
  
Class.forName("com.mysql.jdbc.Driver");  
  
String url = "jdbc:mysql://localhost:3306/jspdb";  
String username = "root";  
String password = "1234";  
  
Connection conn = DriverManager.getConnection(url, username, password);  
Statement sm = conn.createStatement();  
  
int count = sm.executeUpdate(sql);  
if(count == 1){  
    out.println("회원가입 성공!");  
}else{  
    out.println("회원가입 실패!");  
}  
sm.close();  
conn.close();  
%>
```



## [참고] MySQL 드라이버 로딩

### ■ 로딩





## [참고] MySQL 드라이버 로딩

- 일부 주요 데이터베이스들의 `java.sql.Driver`를 구현하는 클래스 이름

Oracle: `oracle.jdbc.driver.OracleDriver`

MySQL: `com.mysql.jdbc.Driver`

Microsoft SQL Server: `com.microsoft.sqlserver.jdbc.SQLServerDriver`

PostgreSQL: `org.postgresql.Driver`

SQLite: `org.sqlite.JDBC`

Apache Derby: `org.apache.derby.jdbc.EmbeddedDriver`

IBM DB2: `com.ibm.db2.jcc.DB2Driver`

MariaDB: `org.mariadb.jdbc.Driver`

:



## DriverManager, the service provider mechanism and memory leaks

`java.sql.DriverManager` supports the [service provider](#) mechanism. This feature is that all the available JDBC drivers that announce themselves by providing a `META-INF/services/java.sql.Driver` file are automatically discovered, loaded and registered, relieving you from the need to load the database driver explicitly before you create a JDBC connection. However, the implementation is fundamentally broken in all Java versions for a servlet container environment. The problem is that `java.sql.DriverManager` will scan for the drivers only once.

The [JRE Memory Leak Prevention Listener](#) that is included with Apache Tomcat solves this by triggering the driver scan during Tomcat startup. This is enabled by default. It means that only libraries visible to the common class loader and its parents will be scanned for database drivers. This include drivers in `$CATALINA_HOME/lib`, `$CATALINA_BASE/lib`, the class path and the module path. Drivers packaged in web applications (in `WEB-INF/lib`) and in the shared class loader (where configured) will not be visible and will not be loaded automatically. If you are considering disabling this feature, note that the scan would be triggered by the first web application that is using JDBC, leading to failures when this web application is reloaded and for other web applications that rely on this feature.

Thus, the web applications that have database drivers in their `WEB-INF/lib` directory cannot rely on the service provider mechanism and should register the drivers explicitly.

The list of drivers in `java.sql.DriverManager` is also a known source of memory leaks. Any Drivers registered by a web application must be deregistered when the web application stops. Tomcat will attempt to automatically discover and deregister any JDBC drivers loaded by the web application class loader when the web application stops. However, it is expected that applications do this for themselves via a `ServletContextListener`.

<https://tomcat.apache.org/tomcat-10.1-doc/jndi-datasource-examples-howto.html>

## [참고] MySQL 드라이버 로딩

- JDBC 드라이버를 로딩하는 두 가지 방법
  1. jdbc.Driver라는 시스템 환경변수에 등록된 내용으로 하는 방법
  2. Class.forName() 메소드를 이용해서 직접 해당 클래스를 로드하는 방법(많이 사용)