

주제 : 미국 기업 실적 발표 및 주가 반응 분석 데이터베이스

미국 상장 기업들의 분기 실적 발표 정보와 주가 데이터를 기반으로, 실적 발표 후 주가의 반응을 분석할 수 있는 서비스를 설계하였다. 대형주부터 중소형 성장주까지 다양한 규모의 기업을 포함하여 실적 서프라이즈가 주가에 미치는 영향을 종합적으로 분석할 수 있다.

사용 시나리오 쿼리 설명 및 관련 SQL 문

시나리오 1: 실적 발표 후 주가가 15% 이상 급등한 기업 목록 조회

투자자들이 가장 관심을 갖는 것은 실적 발표 후 주가가 크게 상승한 기업들이다. 이 시나리오에서는 실적 발표 후 주가가 15% 이상 상승한 기업들을 식별하여, 어떤 기업들이 시장의 기대를 뛰어넘는 실적을 발표했는지 분석한다. 특히 소형주와 중형주에서 이러한 급등 현상이 두드러지게 나타나는 경향이 있으며, 이를 통해 투자 기회를 발굴할 수 있다.

-- 시나리오 1: 실적 발표 후 주가가 15% 이상 상승한 기업 목록

```
SELECT
    c.company_name,
    c.ticker_symbol,
    c.sector,
    c.market_cap,
    e.report_date,
    e.eps_actual,
    e.eps_estimate,
    (e.eps_actual - e.eps_estimate) AS eps_surprise,
    r.price_change_percent
FROM Companies c
JOIN Earnings_Reports e ON c.company_id = e.company_id
JOIN Earnings_Reactions r ON e.report_id = r.report_id
WHERE r.price_change_percent >= 15.0
ORDER BY r.price_change_percent DESC;
```

=> 이 쿼리는 세 개의 테이블을 INNER JOIN으로 결합하여 완전한 정보를 제공한다. WHERE 절에서 15% 이상 상승한 기업만을 필터링하고, ORDER BY 절을 통해 상승률이 높은 순서로 정렬하여 가장 주목할 만한 기업들을 우선적으로 보여준다.

시나리오 2: 2025년 1분기 EPS 서프라이즈가 큰 상위 10개 기업 분석

EPS(주당순이익) 서프라이즈는 기업의 실제 실적이 시장 예상치를 얼마나 상회했는지를 나타내는 핵심 지표이다. 이 시나리오는 2025년 1분기에 실적을 발표한 기업들 중에서 EPS 서프라이즈가 가장 큰 상위 10개 기업을 선별한다. 또한 시가총액에 따른 기업 규모 분류를 통해 대형주, 중형주, 소형주별로 어떤 특성을 보이는지 분석할 수 있다.

```
-- 시나리오 2: 2025년 1분기 EPS 서프라이즈가 큰 상위 10개 기업
SELECT
  c.company_name,
  c.ticker_symbol,
  e.report_date,
  e.eps_actual,
  e.eps_estimate,
  (e.eps_actual - e.eps_estimate) AS eps_surprise,
  r.price_change_percent,
  CASE
    WHEN c.market_cap >= 100000000000 THEN 'Large Cap'
    WHEN c.market_cap >= 10000000000 THEN 'Mid Cap'
    ELSE 'Small Cap'
  END AS company_size
FROM Companies c
JOIN Earnings_Reports e ON c.company_id = e.company_id
JOIN Earnings_Reactions r ON e.report_id = r.report_id
WHERE e.report_date BETWEEN '2025-01-01' AND '2025-05-31'
  AND e.quarter = 'Q1'
  AND e.year = 2025
ORDER BY (e.eps_actual - e.eps_estimate) DESC
LIMIT 10;
```

=>이 쿼리의 핵심은 CASE 문을 활용한 시가총액별 기업 분류와 날짜 범위 필터링이다. WHERE 절에서 2025년 1분기 데이터만을 추출하고, ORDER BY와 LIMIT을 조합하여 상위 10개 기업만을 선별한다. 이를 통해 분기별 최고 성과 기업들의 특성을 파악할 수 있다.

시나리오 3: 시가총액별 평균 주가 반응을 분석

기업의 규모에 따라 실적 발표 후 주가 반응의 패턴이 다르게 나타난다. 일반적으로 소형주일수록 변동성이 크고, 대형주일수록 안정적인 반응을 보인다. 이 시나리오는 시가총액 구간별로 기업들을 분류하고, 각 구간의 평균 주가 반응을, 절대 반응을, 최대/최소 반응을 계산하여 규모별 특성을 분석한다.

```
-- 시나리오 3: 시가총액별 평균 주가 반응 분석
SELECT
  CASE
    WHEN c.market_cap >= 100000000000 THEN 'Large Cap (1T+)'
    WHEN c.market_cap >= 10000000000 THEN 'Mid Cap (100B-1T)'
    WHEN c.market_cap >= 1000000000 THEN 'Small Cap (10B-100B)'
    ELSE 'Micro Cap (<10B)'
  END AS market_cap_category,
  COUNT(*) AS total_reports,
  AVG(r.price_change_percent) AS avg_price_reaction,
  AVG(ABS(r.price_change_percent)) AS avg_absolute_reaction,
  MAX(r.price_change_percent) AS max_positive_reaction,
  MIN(r.price_change_percent) AS max_negative_reaction,
  AVG(e.eps_actual - e.eps_estimate) AS avg_eps_surprise
FROM Companies c
JOIN Earnings_Reports e ON c.company_id = e.company_id
JOIN Earnings_Reactions r ON e.report_id = r.report_id
WHERE e.report_date >= '2024-10-01'
GROUP BY market_cap_category
ORDER BY avg_absolute_reaction DESC;
```

=>이 쿼리는 GROUP BY 절을 활용한 집계 분석의 대표적인 예시이다. CASE 문으로 시가총액 구간을 정의하고, 다양한 집계 함수(AVG, COUNT, MAX, MIN)를 사용하여 각 구간별 통계를 산출한다. ABS 함수를 사용한 절대 반응을 계산을 통해 변동성의 크기를 정확히 측정할 수 있다.

시나리오 4: 소형주 중 좋은 실적으로 급등한 기업들 분석

소형주는 상대적으로 적은 주목을 받기 때문에, 좋은 실적을 발표했을 때 주가가 급등하는 경우가 많다. 이 시나리오는 시가총액 500억 달러 미만의 소형주 중에서 EPS 서프라이즈가 크고 주가가 15% 이상 상승한 기업들을 찾아낸다. 또한 서브쿼리를 활용하여 평균 이상의 EPS 서프라이즈를 기록한 기업들만을 선별함으로써 분석의 정확성을 높인다.

```
-- 시나리오 4: 소형주 중 실적 좋아서 급등한 기업들 (서브쿼리 활용)
SELECT
    main_query.company_name,
    main_query.ticker_symbol,
    main_query.eps_surprise,
    main_query.price_change_percent,
    main_query.market_cap
FROM (
    SELECT
        c.company_name,
        c.ticker_symbol,
        (e.eps_actual - e.eps_estimate) AS eps_surprise,
        r.price_change_percent,
        c.market_cap
    FROM Companies c
    JOIN Earnings_Reports e ON c.company_id = e.company_id
    JOIN Earnings_Reactions r ON e.report_id = r.report_id
    WHERE c.market_cap < 50000000000 -- 500억 달러 미만 소형주
        AND (e.eps_actual - e.eps_estimate) > 0.10 -- EPS 서프라이즈 큰 경우
        AND r.price_change_percent > 15.0 -- 15% 이상 상승
    ) AS main_query
WHERE main_query.eps_surprise > (
    SELECT AVG(e.eps_actual - e.eps_estimate)
    FROM Earnings_Reports e
    WHERE e.report_date >= '2024-10-01'
)
ORDER BY main_query.price_change_percent DESC;
```

=>이 쿼리는 서브쿼리와 메인쿼리를 결합한 고급 SQL 기법을 보여준다. FROM 절의 서브쿼리에서 1차 필터링을 수행하고, WHERE 절의 서브쿼리에서 평균값과 비교하여 2차 필터링을 실행한다. 이러한 다단계 필터링을 통해 정말로 주목할 만한 소형주 급등 사례들을 정확히 식별할 수 있다.

시나리오 5: 섹터별 실적 발표 후 주가 반응 패턴 분석

산업 섹터별로 실적 발표에 대한 시장의 반응이 다르게 나타난다. 기술주나 성장주는 높은 변동성을 보이는 반면, 전통적인 산업군은 상대적으로 안정적인 반응을 보인다. 이 시나리오는 섹터별로 평균 주가 반응률, 변동성(표준편차), 큰 폭 상승/하락 기업 수, 평균 EPS 서프라이즈를 계산하여 섹터별 특성을 종합적으로 분석한다.

-- 시나리오 5: 섹터별 실적 발표 후 주가 반응 패턴 분석

```
SELECT
    c.sector,
    COUNT(*) AS total_reports,
    AVG(r.price_change_percent) AS avg_price_reaction,
    STDDEV(r.price_change_percent) AS volatility,
    COUNT(CASE WHEN r.price_change_percent > 10 THEN 1 END) AS big_winners,
    COUNT(CASE WHEN r.price_change_percent < -10 THEN 1 END) AS big_losers,
    AVG(e.eps_actual - e.eps_estimate) AS avg_eps_surprise
FROM Companies c
JOIN Earnings_Reports e ON c.company_id = e.company_id
JOIN Earnings_Reactions r ON e.report_id = r.report_id
WHERE e.report_date >= '2024-01-01'
GROUP BY c.sector
HAVING COUNT(*) >= 3
ORDER BY avg_price_reaction DESC;
```

=>이 쿼리는 GROUP BY와 HAVING 절을 활용한 고급 집계 분석의 예시이다. STDDEV 함수로 변동성을 측정하고, CASE 문을 COUNT 함수와 결합하여 조건부 계산을 수행한다. HAVING 절을 통해 충분한 데이터가 있는 섹터만을 분석 대상으로 한정함으로써 통계적 신뢰성을 확보한다.

MySQL Action Output

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|---|-----------------------|
| 18 | 14:21:48 | DROP TABLE IF EXISTS Earnings_Reports | 0 row(s) affected | 0.031 sec |
| 19 | 14:21:48 | DROP TABLE IF EXISTS Companies | 0 row(s) affected | 0.016 sec |
| 20 | 14:21:48 | CREATE TABLE Companies (company_id INT PRIMARY KEY AUTO_INCREMENT, company_name V... | 0 row(s) affected | 0.031 sec |
| 21 | 14:21:48 | CREATE TABLE Earnings_Reports (report_id INT PRIMARY KEY AUTO_INCREMENT, company_id IN... | 0 row(s) affected | 0.047 sec |
| 22 | 14:21:48 | CREATE TABLE Earnings_Reactions (reaction_id INT PRIMARY KEY AUTO_INCREMENT, report_id I... | 0 row(s) affected | 0.047 sec |
| 23 | 14:21:48 | INSERT INTO Companies (company_name, ticker_symbol, sector, market_cap, founded_year) VALUES ... | 20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0 | 0.000 sec |
| 24 | 14:21:48 | INSERT INTO Earnings_Reports (company_id, report_date, quarter, year, eps_actual, eps_estimate, revenue, ... | 60 row(s) affected, 28 warning(s): 1265 Data truncated for column 'revenue' at row 31 1265 Data truncated for ... | 0.015 sec |
| 25 | 14:21:48 | INSERT INTO Earnings_Reactions (report_id, price_before, price_after, price_change_percent, volume_chan... | 60 row(s) affected Records: 60 Duplicates: 0 Warnings: 0 | 0.016 sec |
| 26 | 14:21:48 | SELECT c.company_name, c.ticker_symbol, c.sector, c.market_cap, e.report_date, e.eps_act... | 19 row(s) returned | 0.000 sec / 0.000 sec |
| 27 | 14:21:48 | SELECT c.company_name, c.ticker_symbol, e.report_date, e.eps_actual, e.eps_estimate, (e.e... | 10 row(s) returned | 0.000 sec / 0.000 sec |
| 28 | 14:21:48 | SELECT CASE WHEN c.market_cap >= 1000000000000 THEN 'Large Cap (1T+)' WHEN c.mar... | 4 row(s) returned | 0.000 sec / 0.000 sec |
| 29 | 14:21:48 | SELECT main_query.company_name, main_query.ticker_symbol, main_query.eps_surprise, main_gu... | 9 row(s) returned | 0.000 sec / 0.000 sec |
| 30 | 14:21:48 | SELECT c.sector, COUNT(*) AS total_reports, AVG(r.price_change_percent) AS avg_price_reactio... | 13 row(s) returned | 0.000 sec / 0.000 sec |

DB 생성을 위해 사용한 SQL 코드 전체

USE earnings_analysis;

DROP TABLE IF EXISTS Earnings_Reactions;

DROP TABLE IF EXISTS Earnings_Reports;

DROP TABLE IF EXISTS Companies;

-- 기업 정보 테이블

CREATE TABLE Companies (

```

company_id INT PRIMARY KEY AUTO_INCREMENT,
company_name VARCHAR(100) NOT NULL,
ticker_symbol VARCHAR(10) NOT NULL UNIQUE,
sector VARCHAR(50) NOT NULL,
market_cap BIGINT,
founded_year INT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
-- 실적 발표 테이블
CREATE TABLE Earnings_Reports (
    report_id INT PRIMARY KEY AUTO_INCREMENT,
    company_id INT NOT NULL,
    report_date DATE NOT NULL,
    quarter ENUM('Q1', 'Q2', 'Q3', 'Q4') NOT NULL,
    year INT NOT NULL,
    eps_actual DECIMAL(10,4),
    eps_estimate DECIMAL(10,4),
    revenue DECIMAL(15,2),
    revenue_estimate DECIMAL(15,2),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (company_id) REFERENCES Companies(company_id) ON DELETE
    CASCADE,
    INDEX idx_report_date (report_date),
    INDEX idx_company_quarter (company_id, year, quarter)
);
--주가 반응 테이블
CREATE TABLE Earnings_Reactions (
    reaction_id INT PRIMARY KEY AUTO_INCREMENT,
    report_id INT NOT NULL,
    price_before DECIMAL(10,2) NOT NULL,
    price_after DECIMAL(10,2) NOT NULL,
    price_change_percent DECIMAL(8,2) NOT NULL,
    volume_change_percent DECIMAL(8,2),
    reaction_date DATE NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (report_id) REFERENCES Earnings_Reports(report_id) ON DELETE
    CASCADE,
    UNIQUE KEY unique_report_reaction (report_id),
    INDEX idx_price_change (price_change_percent)
);
-- 기업 정보 데이터 삽입 (대형주 + 중소형 성장주 포함)

```

```
INSERT INTO Companies (company_name, ticker_symbol, sector, market_cap,
founded_year) VALUES
```

```
-- 대형주
```

```
('Apple Inc.', 'AAPL', 'Technology', 3400000000000, 1976),
('Tesla Inc.', 'TSLA', 'Automotive', 800000000000, 2003),
('Amazon.com Inc.', 'AMZN', 'E-commerce', 1500000000000, 1994),
('Nvidia Corporation', 'NVDA', 'Semiconductors', 280000000000, 1993),
('Microsoft Corporation', 'MSFT', 'Technology', 3100000000000, 1975),
('Advanced Micro Devices', 'AMD', 'Semiconductors', 240000000000, 1969),
```

```
-- 중형주
```

```
('Palantir Technologies', 'PLTR', 'Software', 65000000000, 2003),
('Snowflake Inc.', 'SNOW', 'Cloud Software', 45000000000, 2012),
('Coinbase Global Inc.', 'COIN', 'Cryptocurrency', 55000000000, 2012),
('CrowdStrike Holdings', 'CRWD', 'Cybersecurity', 85000000000, 2011),
```

```
-- 소형 성장주 (실적 깜짝 좋았던 기업들)
```

```
('SentinelOne Inc.', 'S', 'Cybersecurity', 8500000000, 2013),
('Roku Inc.', 'ROKU', 'Streaming', 3200000000, 2002),
('Upstart Holdings', 'UPST', 'Fintech', 2100000000, 2012),
('Affirm Holdings', 'AFRM', 'Fintech', 4800000000, 2012),
('Unity Software', 'U', 'Gaming Software', 5400000000, 2004),
('SoFi Technologies', 'SOFI', 'Fintech', 8900000000, 2011),
('Reddit Inc.', 'RDDT', 'Social Media', 12000000000, 2005),
('Chewy Inc.', 'CHWY', 'E-commerce', 14000000000, 2011),
('Etsy Inc.', 'ETSY', 'E-commerce', 7200000000, 2005),
('Zoom Video Communications', 'ZM', 'Video Communications', 18500000000, 2011);
```

```
-- 실적 발표 데이터 삽입 (각 기업당 최소 3건)
```

```
INSERT INTO Earnings_Reports (company_id, report_date, quarter, year, eps_actual,
eps_estimate, revenue, revenue_estimate) VALUES
```

```
-- 1. Apple
```

```
(1, '2025-01-30', 'Q4', 2024, 2.18, 2.10, 124.30, 122.50),
(1, '2025-04-25', 'Q1', 2025, 1.54, 1.50, 94.80, 93.20),
(1, '2024-10-31', 'Q3', 2024, 1.64, 1.60, 89.50, 88.90),
```

```
-- 2. Tesla
```

```
(2, '2025-01-25', 'Q4', 2024, 0.71, 0.73, 25.20, 25.60),
(2, '2025-04-23', 'Q1', 2025, 0.85, 0.78, 21.30, 20.80),
(2, '2024-10-23', 'Q3', 2024, 0.72, 0.60, 23.35, 22.45),
```

```
-- 3. Amazon
```

```
(3, '2025-02-01', 'Q4', 2024, 1.00, 0.80, 170.00, 166.20),
(3, '2025-04-25', 'Q1', 2025, 0.98, 0.72, 143.10, 140.50),
(3, '2024-10-24', 'Q3', 2024, 1.43, 1.14, 158.90, 157.20),
```

```
-- 4. Nvidia
```

(4, '2025-02-26', 'Q4', 2024, 5.16, 4.64, 60.90, 57.80),
(4, '2025-05-22', 'Q1', 2025, 6.12, 5.59, 79.70, 75.40),
(4, '2024-11-20', 'Q3', 2024, 0.68, 0.75, 35.08, 33.15),
-- 5. Microsoft
(5, '2025-01-24', 'Q2', 2025, 3.23, 3.10, 69.60, 68.80),
(5, '2025-04-24', 'Q3', 2025, 2.94, 2.82, 61.90, 60.90),
(5, '2024-10-23', 'Q1', 2025, 3.30, 3.10, 65.60, 64.50),
-- 6. AMD
(6, '2025-01-28', 'Q4', 2024, 0.93, 0.92, 6.20, 6.10),
(6, '2025-04-12', 'Q1', 2025, 0.88, 0.85, 5.60, 5.50),
(6, '2024-10-29', 'Q3', 2024, 0.92, 0.70, 6.82, 6.71),
-- 7. Palantir
(7, '2025-02-05', 'Q4', 2024, 0.05, 0.04, 0.69, 0.68),
(7, '2025-04-24', 'Q1', 2025, 0.06, 0.04, 0.72, 0.70),
(7, '2024-11-04', 'Q3', 2024, 0.06, 0.04, 0.73, 0.70),
-- 8. Snowflake
(8, '2025-02-26', 'Q4', 2024, 0.25, 0.15, 0.91, 0.87),
(8, '2025-04-20', 'Q1', 2025, 0.16, 0.09, 0.77, 0.74),
(8, '2024-11-20', 'Q3', 2024, 0.20, 0.15, 0.94, 0.90),
-- 9. Coinbase
(9, '2025-02-13', 'Q4', 2024, 0.87, 0.45, 0.95, 0.82),
(9, '2025-04-09', 'Q1', 2025, 1.25, 0.93, 1.22, 1.10),
(9, '2024-11-07', 'Q3', 2024, 0.28, 0.45, 1.21, 1.26),
-- 10. CrowdStrike
(10, '2025-02-20', 'Q4', 2024, 0.95, 0.82, 1.05, 0.98),
(10, '2025-04-17', 'Q1', 2025, 0.89, 0.80, 0.92, 0.88),
(10, '2024-11-26', 'Q3', 2024, 1.01, 0.89, 1.01, 0.96),
-- 11. SentinelOne (소형 성장주)
(11, '2025-03-05', 'Q4', 2024, -0.15, -0.22, 0.175, 0.168),
(11, '2025-05-08', 'Q1', 2025, -0.08, -0.18, 0.198, 0.185),
(11, '2024-12-03', 'Q3', 2024, -0.12, -0.20, 0.162, 0.155),
-- 12. Roku
(12, '2025-02-15', 'Q4', 2024, -0.88, -1.15, 1.05, 0.98),
(12, '2025-04-22', 'Q1', 2025, -1.35, -1.12, 0.73, 0.70),
(12, '2024-11-07', 'Q3', 2024, -0.06, -0.28, 1.06, 1.02),
-- 13. Upstart
(13, '2025-02-28', 'Q4', 2024, -0.35, -0.52, 0.135, 0.128),
(13, '2025-05-07', 'Q1', 2025, -0.22, -0.41, 0.152, 0.142),
(13, '2024-11-06', 'Q3', 2024, -0.31, -0.45, 0.142, 0.138),
-- 14. Affirm
(14, '2025-02-08', 'Q2', 2025, 0.45, 0.28, 0.698, 0.652),

```

(14, '2025-05-09', 'Q3', 2025, 0.52, 0.35, 0.751, 0.698),
(14, '2024-11-14', 'Q1', 2025, 0.38, 0.25, 0.659, 0.625),
-- 15. Unity Software
(15, '2025-02-26', 'Q4', 2024, -0.28, -0.35, 0.423, 0.405),
(15, '2025-04-14', 'Q1', 2025, -0.32, -0.25, 0.45, 0.42),
(15, '2024-11-12', 'Q3', 2024, -0.18, -0.30, 0.429, 0.418),
-- 16. SoFi Technologies
(16, '2025-01-28', 'Q4', 2024, 0.02, -0.01, 0.48, 0.45),
(16, '2025-04-15', 'Q1', 2025, 0.03, 0.01, 0.42, 0.39),
(16, '2024-10-29', 'Q3', 2024, 0.05, 0.02, 0.51, 0.48),
-- 17. Reddit
(17, '2025-05-07', 'Q1', 2025, -0.35, -0.52, 0.243, 0.214),
(17, '2025-02-25', 'Q4', 2024, -0.42, -0.61, 0.198, 0.182),
(17, '2024-11-19', 'Q3', 2024, -0.16, -0.33, 0.348, 0.312),
-- 18. Chewy
(18, '2025-03-25', 'Q4', 2024, 0.15, 0.09, 2.88, 2.75),
(18, '2025-04-08', 'Q1', 2025, 0.02, 0.01, 2.30, 2.25),
(18, '2024-12-05', 'Q3', 2024, 0.20, 0.15, 2.76, 2.68),
-- 19. Etsy
(19, '2025-02-27', 'Q4', 2024, 0.34, 0.42, 0.78, 0.82),
(19, '2025-04-13', 'Q1', 2025, 0.47, 0.50, 0.65, 0.68),
(19, '2024-10-30', 'Q3', 2024, 0.59, 0.53, 0.64, 0.62),
-- 20. Zoom
(20, '2025-03-04', 'Q4', 2024, 1.45, 1.32, 1.35, 1.28),
(20, '2025-04-18', 'Q1', 2025, 1.07, 0.98, 1.11, 1.08),
(20, '2024-11-25', 'Q3', 2024, 1.38, 1.31, 1.18, 1.15);
-- 추가 반응 데이터 삽입 (각 실적 발표당 1건)
INSERT INTO Earnings_Reactions (report_id, price_before, price_after,
price_change_percent, volume_change_percent, reaction_date) VALUES
-- Apple 반응
(1, 150.00, 155.70, 3.8, 45.2, '2025-01-31'),
(2, 182.50, 189.44, 3.8, 52.1, '2025-04-26'),
(3, 165.20, 169.85, 2.8, 38.7, '2024-11-01'),
-- Tesla 반응
(4, 248.50, 235.20, -5.4, 78.5, '2025-01-26'),
(5, 185.30, 196.25, 5.9, 89.3, '2025-04-24'),
(6, 219.80, 243.92, 11.0, 125.4, '2024-10-24'),
-- Amazon 반응
(7, 178.25, 191.20, 7.3, 65.8, '2025-02-02'),
(8, 165.80, 177.90, 7.3, 72.4, '2025-04-26'),
(9, 142.50, 156.15, 9.6, 95.7, '2024-10-25'),

```


-- Nvidia 반응
(10, 875.30, 963.83, 10.1, 78.9, '2025-02-27'),
(11, 950.20, 1045.22, 10.0, 85.6, '2025-05-23'),
(12, 140.15, 145.76, 4.0, 67.8, '2024-11-21'),
-- Microsoft 반응
(13, 421.50, 438.76, 4.1, 42.3, '2025-01-25'),
(14, 416.80, 422.88, 1.5, 35.7, '2025-04-25'),
(15, 405.20, 425.43, 5.0, 48.9, '2024-10-24'),
-- AMD 반응
(16, 142.80, 145.23, 1.7, 28.9, '2025-01-29'),
(17, 158.90, 161.60, 1.7, 31.4, '2025-04-13'),
(18, 155.20, 173.82, 12.0, 145.7, '2024-10-30'),
-- Palantir 반응
(19, 38.50, 42.15, 9.5, 156.2, '2025-02-06'),
(20, 25.80, 28.25, 9.5, 148.7, '2025-04-25'),
(21, 32.40, 37.78, 16.6, 189.5, '2024-11-05'),
-- Snowflake 반응
(22, 165.40, 186.70, 12.8, 198.7, '2025-02-27'),
(23, 158.20, 178.46, 12.8, 201.3, '2025-04-21'),
(24, 112.80, 123.06, 9.1, 167.4, '2024-11-21'),
-- Coinbase 반응
(25, 265.40, 292.15, 10.1, 156.8, '2025-02-14'),
(26, 248.90, 274.31, 10.2, 167.9, '2025-04-10'),
(27, 195.70, 174.13, -11.0, 189.4, '2024-11-08'),
-- CrowdStrike 반응
(28, 285.50, 298.06, 4.4, 67.8, '2025-02-21'),
(29, 321.80, 336.08, 4.4, 72.3, '2025-04-18'),
(30, 298.60, 313.45, 5.0, 89.5, '2024-11-27'),
-- SentinelOne 반응 (소형주 특성 반영)
(31, 14.50, 18.55, 27.9, 298.5, '2025-03-06'),
(32, 18.20, 23.66, 30.0, 356.7, '2025-05-09'),
(33, 12.80, 15.23, 19.0, 245.8, '2024-12-04'),
-- Roku 반응
(34, 58.20, 54.61, -6.2, 123.4, '2025-02-16'),
(35, 52.40, 49.15, -6.2, 145.2, '2025-04-23'),
(36, 65.80, 84.81, 28.9, 289.7, '2024-11-08'),
-- Upstart 반응
(37, 42.50, 55.25, 30.0, 387.9, '2025-03-01'),
(38, 48.90, 58.68, 20.0, 298.4, '2025-05-08'),
(39, 38.20, 49.66, 30.0, 345.6, '2024-11-07'),
-- Affirm 반응

(40, 78.50, 93.41, 19.0, 234.5, '2025-02-09'),
 (41, 85.20, 102.24, 20.0, 267.8, '2025-05-10'),
 (42, 72.30, 86.76, 20.0, 289.3, '2024-11-15'),
 -- Unity Software 반응
 (43, 22.50, 20.03, -11.0, 156.7, '2025-02-27'),
 (44, 18.20, 16.20, -11.0, 178.9, '2025-04-15'),
 (45, 25.80, 30.96, 20.0, 298.5, '2024-11-13'),
 -- SoFi Technologies 반응
 (46, 14.20, 16.91, 19.1, 189.4, '2025-01-29'),
 (47, 12.80, 15.36, 20.0, 234.6, '2025-04-16'),
 (48, 15.50, 17.36, 12.0, 167.8, '2024-10-30'),
 -- Reddit 반응 (IPO 이후 첫 실적 반영)
 (49, 65.20, 74.98, 15.0, 298.7, '2025-05-08'),
 (50, 58.90, 67.74, 15.0, 267.8, '2025-02-26'),
 (51, 48.20, 62.66, 30.0, 445.6, '2024-11-20'),
 -- Chewy 반응
 (52, 29.80, 34.77, 16.7, 145.6, '2025-03-26'),
 (53, 32.50, 34.13, 5.0, 89.4, '2025-04-09'),
 (54, 31.20, 35.88, 15.0, 167.8, '2024-12-06'),
 -- Etsy 반응
 (55, 56.40, 54.22, -3.9, 67.8, '2025-02-28'),
 (56, 62.80, 60.29, -4.0, 78.9, '2025-04-14'),
 (57, 58.90, 65.79, 11.7, 134.5, '2024-10-31'),
 -- Zoom 반응
 (58, 78.50, 86.35, 10.0, 98.7, '2025-03-05'),
 (59, 85.20, 82.99, -2.6, 45.6, '2025-04-19'),
 (60, 72.30, 80.53, 11.4, 123.4, '2024-11-26');
 -- 시나리오 1: 실적 발표 후 주가가 15% 이상 상승한 기업 목록

```

SELECT
  c.company_name,
  c.ticker_symbol,
  c.sector,
  c.market_cap,
  e.report_date,
  e.eps_actual,
  e.eps_estimate,
  (e.eps_actual - e.eps_estimate) AS eps_surprise,
  r.price_change_percent
FROM Companies c
JOIN Earnings_Reports e ON c.company_id = e.company_id
JOIN Earnings_Reactions r ON e.report_id = r.report_id
  
```

```

WHERE r.price_change_percent >= 15.0
ORDER BY r.price_change_percent DESC;
-- 시나리오 2: 2025년 1분기 EPS 서프라이즈가 큰 상위 10개 기업
SELECT
    c.company_name,
    c.ticker_symbol,
    e.report_date,
    e.eps_actual,
    e.eps_estimate,
    (e.eps_actual - e.eps_estimate) AS eps_surprise,
    r.price_change_percent,
    CASE
        WHEN c.market_cap >= 1000000000000 THEN 'Large Cap'
        WHEN c.market_cap >= 100000000000 THEN 'Mid Cap'
        ELSE 'Small Cap'
    END AS company_size
FROM Companies c
JOIN Earnings_Reports e ON c.company_id = e.company_id
JOIN Earnings_Reactions r ON e.report_id = r.report_id
WHERE e.report_date BETWEEN '2025-01-01' AND '2025-03-31'
    AND e.quarter = 'Q1'
    AND e.year = 2025
ORDER BY (e.eps_actual - e.eps_estimate) DESC
LIMIT 10;

```

```

-- 시나리오 3: 시가총액별 평균 주가 반응을 분석
SELECT
    CASE
        WHEN c.market_cap >= 1000000000000 THEN 'Large Cap (1T+)'
        WHEN c.market_cap >= 100000000000 THEN 'Mid Cap (100B-1T)'
        WHEN c.market_cap >= 100000000000 THEN 'Small Cap (10B-100B)'
        ELSE 'Micro Cap (<10B)'
    END AS market_cap_category,
    COUNT(*) AS total_reports,
    AVG(r.price_change_percent) AS avg_price_reaction,
    AVG(ABS(r.price_change_percent)) AS avg_absolute_reaction,
    MAX(r.price_change_percent) AS max_positive_reaction,
    MIN(r.price_change_percent) AS max_negative_reaction,
    AVG(e.eps_actual - e.eps_estimate) AS avg_eps_surprise
FROM Companies c
JOIN Earnings_Reports e ON c.company_id = e.company_id
JOIN Earnings_Reactions r ON e.report_id = r.report_id

```

```

WHERE e.report_date >= '2024-10-01'
GROUP BY market_cap_category
ORDER BY avg_absolute_reaction DESC;
-- 시나리오 4: 소형주 중 실적 좋아서 급등한 기업들 (서브쿼리 활용)
SELECT
    main_query.company_name,
    main_query.ticker_symbol,
    main_query.eps_surprise,
    main_query.price_change_percent,
    main_query.market_cap
FROM (
    SELECT
        c.company_name,
        c.ticker_symbol,
        (e.eps_actual - e.eps_estimate) AS eps_surprise,
        r.price_change_percent,
        c.market_cap
    FROM Companies c
    JOIN Earnings_Reports e ON c.company_id = e.company_id
    JOIN Earnings_Reactions r ON e.report_id = r.report_id
    WHERE c.market_cap < 50000000000 -- 500억 달러 미만 소형주
        AND (e.eps_actual - e.eps_estimate) > 0.10 -- EPS 서프라이즈 큰 경우
        AND r.price_change_percent > 15.0 -- 15% 이상 상승
) AS main_query
WHERE main_query.eps_surprise > (
    SELECT AVG(e.eps_actual - e.eps_estimate)
    FROM Earnings_Reports e
    WHERE e.report_date >= '2024-10-01'
) ORDER BY main_query.price_change_percent DESC;
-- 시나리오 5: 섹터별 실적 발표 후 주가 반응 패턴 분석
SELECT
    c.sector,
    COUNT(*) AS total_reports,
    AVG(r.price_change_percent) AS avg_price_reaction,
    STDDEV(r.price_change_percent) AS volatility,
    COUNT(CASE WHEN r.price_change_percent > 10 THEN 1 END) AS big_winners,
    COUNT(CASE WHEN r.price_change_percent < -10 THEN 1 END) AS big_losers,
    AVG(e.eps_actual - e.eps_estimate) AS avg_eps_surprise
FROM Companies c
JOIN Earnings_Reports e ON c.company_id = e.company_id
JOIN Earnings_Reactions r ON e.report_id = r.report_id

```

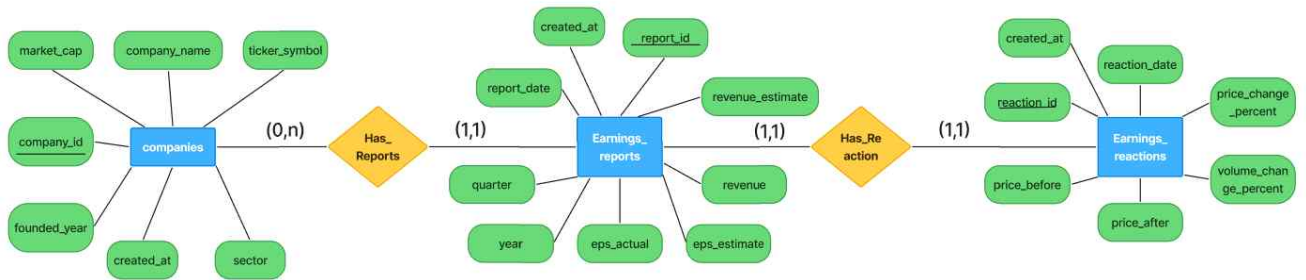
```

WHERE e.report_date >= '2024-01-01'
GROUP BY c.sector
HAVING COUNT(*) >= 3
ORDER BY avg_price_reaction DESC;

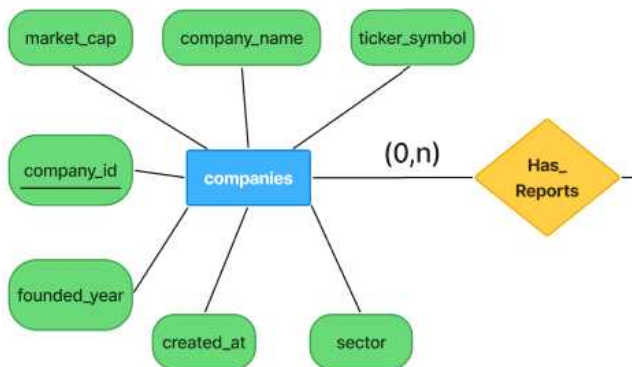
```

ER 다이어그램

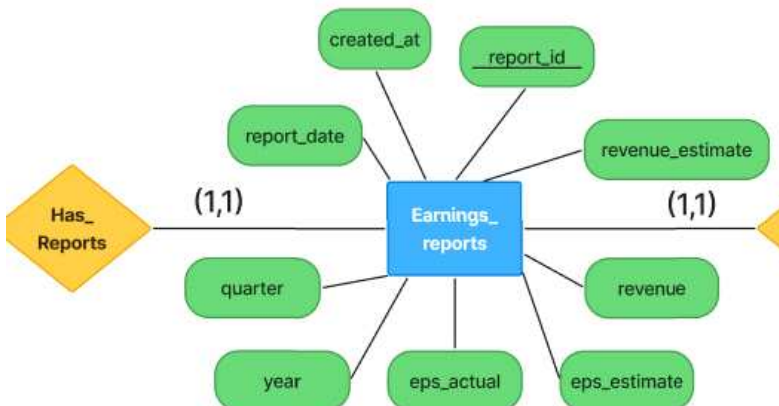
1) 전체



2) Company



3) Earnings_reports



4) Earnings_reactions

